US009189280B2

US009189280B2

(12) **United States Patent**
Park et al.

(10) **Patent No.:** **US 9,189,280 B2**
(45) **Date of Patent:** **Nov. 17, 2015**

(54) **TRACKING LARGE NUMBERS OF MOVING OBJECTS IN AN EVENT PROCESSING SYSTEM**

(75) Inventors: **Hoyong Park**, San Jose, CA (US); **Eric Hsiao**, San Mateo, CA (US); **Andrew Piper**, Amersham (GB)

(73) Assignee: **Oracle International Corporation**, Redwood Shores, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **13/107,742**

(22) Filed: **May 13, 2011**

(51) **Int. Cl.**
G06F 9/46	(2006.01)
G06F 9/50	(2006.01)
G06F 17/30	(2006.01)

(52) **U.S. Cl.**
CPC *G06F 9/5027* (2013.01); *G06F 17/30324* (2013.01)

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,996,687	A	2/1991	Hess et al.
5,051,947	A	9/1991	Messenger et al.
5,339,392	A	8/1994	Risberg et al.
5,495,600	A	2/1996	Terry et al.
5,706,494	A	1/1998	Cochrane et al.
5,802,262	A	9/1998	Van De Vanter
5,802,523	A	9/1998	Jasuja et al.
5,822,750	A	10/1998	Jou et al.
5,826,077	A	10/1998	Blakeley et al.
5,850,544	A	12/1998	Parvathaneny et al.
5,857,182	A	1/1999	DeMichiel et al.
5,918,225	A	6/1999	White et al.
5,920,716	A	7/1999	Johnson et al.
5,937,195	A	8/1999	Ju et al.

(Continued)

FOREIGN PATENT DOCUMENTS

EP	1241589 A2	9/2002
EP	2474922	7/2012

(Continued)

OTHER PUBLICATIONS

"Oracle Complex Event Processing CQL Language Reference," 11g Release 1 (11.1.1) E12048-01, Apr. 2010, 540 pages.

(Continued)

Primary Examiner — H S Sough
Assistant Examiner — Umut Onat
(74) *Attorney, Agent, or Firm* — Kilpatrick Townsend and Stockton LLP

(57) **ABSTRACT**

Techniques for tracking large numbers of moving objects in an event processing system are provided. An input event stream can be received, where the events in the input event stream represent the movement of a plurality of geometries or objects. The input event stream can then be partitioned among a number of processing nodes of the event processing system, thereby enabling parallel processing of one or more continuous queries for tracking the objects. The partitioning can be performed such that each processing node is configured to track objects in a predefined spatial region, and the spatial regions for at least two nodes overlap. This overlapping window enables a single node to find, e.g., all of the objects within a particular distance of a target object, even if the target object is in the process of moving from the region of that node to the overlapping region of another node.
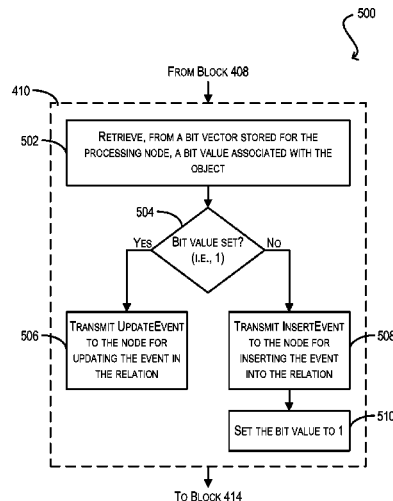
20 Claims, 8 Drawing Sheets

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,937,401	A	8/1999	Hillegas
6,006,235	A	12/1999	Macdonald et al.
6,011,916	A	1/2000	Moore et al.
6,041,344	A	3/2000	Bodamer et al.
6,081,801	A	6/2000	Cochrane et al.
6,092,065	A	7/2000	Floratos et al.
6,108,666	A	8/2000	Floratos et al.
6,112,198	A	8/2000	Lohman et al.
6,128,610	A	10/2000	Srinivasan et al.
6,158,045	A	12/2000	You
6,219,660	B1	4/2001	Haderle et al.
6,263,332	B1	7/2001	Nasr et al.
6,278,994	B1	8/2001	Fuh et al.
6,282,537	B1	8/2001	Madnick et al.
6,341,281	B1	1/2002	MacNicol et al.
6,353,821	B1	3/2002	Gray
6,367,034	B1	4/2002	Novik et al.
6,370,537	B1	4/2002	Gilbert et al.
6,389,436	B1	5/2002	Chakrabarti et al.
6,397,262	B1 *	5/2002	Hayden et al. 719/318
6,418,448	B1	7/2002	Sarkar
6,438,540	B2	8/2002	Nasr et al.
6,438,559	B1	8/2002	White et al.
6,439,783	B1	8/2002	Antoshenkov
6,449,620	B1	9/2002	Draper et al.
6,453,314	B1	9/2002	Chan et al.
6,507,834	B1	1/2003	Kabra et al.
6,523,102	B1	2/2003	Dye et al.
6,546,381	B1	4/2003	Subramanian et al.
6,615,203	B1	9/2003	Lin et al.
6,681,343	B1	1/2004	Nakabo
6,708,186	B1	3/2004	Claborn et al.
6,718,278	B1 *	4/2004	Steggles 702/150
6,748,386	B1	6/2004	Li
6,751,619	B1	6/2004	Rowstron et al.
6,766,330	B1	7/2004	Chen et al.
6,785,677	B1	8/2004	Fritchman
6,826,566	B2	11/2004	Lewak et al.
6,836,778	B2	12/2004	Manikutty et al.
6,850,925	B2	2/2005	Chaudhuri et al.
6,856,981	B2	2/2005	Wyschogrod et al.
6,985,904	B1	1/2006	Kaluskar et al.
6,996,557	B1	2/2006	Leung et al.
7,020,696	B1	3/2006	Perry et al.
7,047,249	B1	5/2006	Vincent
7,051,034	B1	5/2006	Ghosh et al.
7,062,749	B2	6/2006	Cyr et al.
7,080,062	B1	7/2006	Leung et al.
7,093,023	B2	8/2006	Lockwood et al.
7,145,938	B2	12/2006	Takeuchi et al.
7,146,352	B2	12/2006	Brundage et al.
7,167,848	B2	1/2007	Boukouvalas et al.
7,203,927	B2	4/2007	Al-Azzawe et al.
7,224,185	B2	5/2007	Campbell et al.
7,225,188	B1	5/2007	Gai et al.
7,236,972	B2	6/2007	Lewak et al.
7,305,391	B2	12/2007	Wyschogrod et al.
7,308,561	B2	12/2007	Cornet et al.
7,310,638	B1	12/2007	Blair
7,376,656	B2	5/2008	Blakeley et al.
7,383,253	B1	6/2008	Tsimelzon et al.
7,403,959	B2	7/2008	Nishizawa et al.
7,430,549	B2	9/2008	Zane et al.
7,451,143	B2	11/2008	Sharangpani et al.
7,475,058	B2	1/2009	Kakivaya et al.
7,483,976	B2	1/2009	Ross
7,516,121	B2	4/2009	Liu et al.
7,519,577	B2	4/2009	Brundage et al.
7,519,962	B2	4/2009	Aman
7,533,087	B2	5/2009	Liu et al.
7,546,284	B1	6/2009	Martinez et al.
7,552,365	B1	6/2009	Marsh et al.
7,567,953	B2	7/2009	Kadayam et al.
7,580,946	B2	8/2009	Mansour et al.
7,587,383	B2	9/2009	Koo et al.
7,603,674	B2	10/2009	Cyr et al.
7,613,848	B2	11/2009	Amini et al.
7,620,851	B1	11/2009	Leavy et al.
7,630,982	B2	12/2009	Boyce
7,634,501	B2	12/2009	Yabloko
7,636,703	B2	12/2009	Taylor
7,644,066	B2	1/2010	Krishnaprasad et al.
7,653,645	B1	1/2010	Stokes
7,672,964	B1	3/2010	Yan et al.
7,673,065	B2	3/2010	Srinivasan et al.
7,676,461	B2	3/2010	Chkodrov et al.
7,689,622	B2	3/2010	Liu et al.
7,693,891	B2	4/2010	Stokes et al.
7,702,629	B2	4/2010	Cytron et al.
7,702,639	B2	4/2010	Stanley et al.
7,711,782	B2	5/2010	Kim et al.
7,716,210	B2	5/2010	Ozcan et al.
7,739,265	B2	6/2010	Jain et al.
7,805,445	B2	9/2010	Boyer et al.
7,814,111	B2	10/2010	Levin
7,818,313	B2	10/2010	Tsimelzon
7,823,066	B1	10/2010	Kuramura
7,827,146	B1	11/2010	De Landstheer et al.
7,827,190	B2	11/2010	Pandya
7,844,829	B2	11/2010	Meenakshisundaram
7,870,124	B2	1/2011	Liu et al.
7,877,381	B2	1/2011	Ewen et al.
7,895,187	B2	2/2011	Bowman
7,912,853	B2	3/2011	Agrawal
7,917,299	B2	3/2011	Buhler et al.
7,930,322	B2	4/2011	MacLennan
7,945,540	B2	5/2011	Park et al.
7,953,728	B2	5/2011	Hu et al.
7,954,109	B1	5/2011	Durham et al.
7,979,420	B2	7/2011	Jain et al.
7,987,204	B2	7/2011	Stokes
7,988,817	B2	8/2011	Son
7,991,766	B2	8/2011	Srinivasan et al.
7,996,388	B2	8/2011	Jain et al.
8,019,747	B2	9/2011	Srinivasan et al.
8,032,544	B2	10/2011	Jing et al.
8,046,747	B2	10/2011	Cyr et al.
8,073,826	B2	12/2011	Srinivasan et al.
8,099,400	B2	1/2012	Haub et al.
8,103,655	B2	1/2012	Srinivasan et al.
8,134,184	B2	3/2012	Becker et al.
8,155,880	B2 *	4/2012	Patel et al. 701/400
8,195,648	B2	6/2012	Zabback et al.
8,204,873	B2	6/2012	Chavan
8,204,875	B2	6/2012	Srinivasan et al.
8,260,803	B2	9/2012	Hsu et al.
8,290,776	B2	10/2012	Moriwaki et al.
8,296,316	B2	10/2012	Jain et al.
8,315,990	B2	11/2012	Barga et al.
8,316,012	B2	11/2012	Abouzied et al.
8,346,511	B2	1/2013	Schoning et al.
8,370,812	B2	2/2013	Feblowitz et al.
8,392,402	B2	3/2013	Mihaila et al.
8,447,744	B2	5/2013	Alves et al.
8,458,175	B2	6/2013	Stokes
8,498,956	B2	7/2013	Srinivasan et al.
8,521,867	B2	8/2013	Srinivasan et al.
8,527,458	B2	9/2013	Park et al.
8,543,558	B2	9/2013	Srinivasan et al.
8,572,589	B2	10/2013	Cataldo et al.
8,589,436	B2	11/2013	Srinivasan et al.
8,676,841	B2	3/2014	Srinivasan et al.
8,713,049	B2	4/2014	Jain et al.
8,745,070	B2	6/2014	Krisnamurthy
8,762,369	B2	6/2014	Macho et al.
8,775,412	B2	7/2014	Day et al.
2002/0023211	A1	2/2002	Roth et al.
2002/0032804	A1	3/2002	Hunt
2002/0038313	A1	3/2002	Klein et al.
2002/0049788	A1	4/2002	Lipkin et al.
2002/0056004	A1	5/2002	Smith et al.
2002/0116362	A1	8/2002	Li et al.
2002/0116371	A1	8/2002	Dodds et al.
2002/0133484	A1	9/2002	Chau et al.

(56) **References Cited**

U.S. PATENT DOCUMENTS

2002/0169788 A1	11/2002	Lee et al.
2003/0014408 A1	1/2003	Robertson
2003/0037048 A1	2/2003	Kabra et al.
2003/0046673 A1	3/2003	Copeland et al.
2003/0065655 A1	4/2003	Syeda-Mahmood
2003/0065659 A1	4/2003	Agarwal et al.
2003/0120682 A1	6/2003	Bestgen et al.
2003/0135304 A1	7/2003	Sroub et al.
2003/0200198 A1	10/2003	Chandrasekar et al.
2003/0229652 A1	12/2003	Bakalash et al.
2003/0236766 A1	12/2003	Fortuna et al.
2004/0010496 A1	1/2004	Behrendt et al.
2004/0019592 A1	1/2004	Crabtree
2004/0024773 A1	2/2004	Stoffel et al.
2004/0064466 A1	4/2004	Manikutty et al.
2004/0073534 A1	4/2004	Robson
2004/0088404 A1	5/2004	Aggarwal
2004/0117359 A1	6/2004	Snodgrass et al.
2004/0136598 A1	7/2004	Le Leannec et al.
2004/0151382 A1	8/2004	Stellenberg et al.
2004/0153329 A1	8/2004	Casati et al.
2004/0167864 A1	8/2004	Wang et al.
2004/0168107 A1	8/2004	Sharp et al.
2004/0177053 A1	9/2004	Donoho et al.
2004/0201612 A1	10/2004	Hild et al.
2004/0205082 A1	10/2004	Fontoura et al.
2004/0220896 A1	11/2004	Finlay et al.
2004/0220912 A1	11/2004	Manikutty et al.
2004/0220927 A1	11/2004	Murthy et al.
2004/0243590 A1	12/2004	Gu et al.
2004/0267760 A1	12/2004	Brundage et al.
2004/0268314 A1	12/2004	Kollman et al.
2005/0010896 A1	1/2005	Meliksetian et al.
2005/0055338 A1	3/2005	Warner et al.
2005/0065949 A1	3/2005	Warner et al.
2005/0096124 A1	5/2005	Stronach
2005/0097128 A1	5/2005	Ryan et al.
2005/0120016 A1	6/2005	Midgley
2005/0154740 A1	7/2005	Day et al.
2005/0174940 A1	8/2005	Iny
2005/0177579 A1	8/2005	Blakeley et al.
2005/0192921 A1	9/2005	Chaudhuri et al.
2005/0204340 A1	9/2005	Ruminer et al.
2005/0229158 A1	10/2005	Thusoo et al.
2005/0273352 A1	12/2005	Moffat et al.
2005/0273450 A1	12/2005	McMillen et al.
2005/0289125 A1	12/2005	Liu et al.
2006/0007308 A1*	1/2006	Ide et al. 348/143
2006/0015482 A1	1/2006	Beyer et al.
2006/0031204 A1	2/2006	Liu et al.
2006/0047696 A1	3/2006	Larson et al.
2006/0064487 A1	3/2006	Ross
2006/0080646 A1	4/2006	Aman
2006/0085592 A1	4/2006	Ganguly et al.
2006/0089939 A1	4/2006	Broda et al.
2006/0100957 A1	5/2006	Buttler et al.
2006/0100969 A1	5/2006	Wang et al.
2006/0106786 A1	5/2006	Day et al.
2006/0106797 A1	5/2006	Srinivasa et al.
2006/0129554 A1	6/2006	Suyama et al.
2006/0155719 A1	7/2006	Mihaeli et al.
2006/0167704 A1	7/2006	Nicholls et al.
2006/0167856 A1	7/2006	Angele et al.
2006/0212441 A1	9/2006	Tang et al.
2006/0224576 A1	10/2006	Liu et al.
2006/0230029 A1	10/2006	Yan
2006/0235840 A1	10/2006	Manikutty et al.
2006/0242180 A1	10/2006	Graf et al.
2006/0282429 A1	12/2006	Hernandez-Sherrington et al.
2006/0294095 A1	12/2006	Berk et al.
2007/0016467 A1	1/2007	John et al.
2007/0022092 A1	1/2007	Nishizawa et al.
2007/0039049 A1	2/2007	Kupferman et al.
2007/0050340 A1	3/2007	von Kaenel et al.
2007/0076314 A1	4/2007	Rigney
2007/0118600 A1	5/2007	Arora
2007/0136239 A1	6/2007	Lee et al.
2007/0136254 A1	6/2007	Choi et al.
2007/0156787 A1	7/2007	MacGregor
2007/0156964 A1*	7/2007	Sistla 711/133
2007/0192301 A1*	8/2007	Posner 707/3
2007/0198479 A1	8/2007	Cai et al.
2007/0226188 A1	9/2007	Johnson et al.
2007/0226239 A1	9/2007	Johnson et al.
2007/0271280 A1	11/2007	Chandasekaran
2007/0294217 A1	12/2007	Chen et al.
2008/0005093 A1	1/2008	Liu et al.
2008/0010093 A1	1/2008	LaPlante et al.
2008/0010241 A1	1/2008	McGoveran
2008/0016095 A1	1/2008	Bhatnagar et al.
2008/0028095 A1	1/2008	Lang et al.
2008/0033914 A1	2/2008	Cherniack et al.
2008/0034427 A1	2/2008	Cadambi et al.
2008/0046401 A1	2/2008	Lee et al.
2008/0071904 A1	3/2008	Schuba et al.
2008/0077570 A1	3/2008	Tang et al.
2008/0077587 A1	3/2008	Wyschogrod et al.
2008/0077780 A1	3/2008	Zingher
2008/0082484 A1	4/2008	Averbuch et al.
2008/0082514 A1	4/2008	Khorlin et al.
2008/0086321 A1	4/2008	Walton
2008/0098359 A1	4/2008	Ivanov et al.
2008/0110397 A1	5/2008	Son
2008/0114787 A1	5/2008	Kashiyama et al.
2008/0120283 A1	5/2008	Liu et al.
2008/0162583 A1	7/2008	Brown et al.
2008/0195577 A1	8/2008	Fan et al.
2008/0235298 A1	9/2008	Lin et al.
2008/0243451 A1	10/2008	Feblowitz et al.
2008/0243675 A1*	10/2008	Parsons et al. 705/37
2008/0250073 A1	10/2008	Nori et al.
2008/0255847 A1	10/2008	Moriwaki et al.
2008/0263039 A1	10/2008	Van Lunteren
2008/0270764 A1	10/2008	McMillen et al.
2008/0281782 A1	11/2008	Agrawal
2008/0301124 A1	12/2008	Alves et al.
2008/0301125 A1	12/2008	Alves et al.
2008/0301135 A1	12/2008	Alves et al.
2008/0301256 A1	12/2008	McWilliams et al.
2008/0313131 A1	12/2008	Friedman et al.
2009/0006320 A1	1/2009	Ding et al.
2009/0006346 A1	1/2009	C N et al.
2009/0007098 A1	1/2009	Chevrette et al.
2009/0019045 A1	1/2009	Amir et al.
2009/0024622 A1	1/2009	Chkodrov et al.
2009/0043729 A1	2/2009	Liu et al.
2009/0070355 A1	3/2009	Cadarette et al.
2009/0070785 A1	3/2009	Alvez et al.
2009/0070786 A1	3/2009	Alves et al.
2009/0076899 A1	3/2009	Gbodimowo
2009/0088962 A1*	4/2009	Jones 701/200
2009/0100029 A1	4/2009	Jain et al.
2009/0106189 A1	4/2009	Jain et al.
2009/0106190 A1	4/2009	Srinivasan et al.
2009/0106198 A1	4/2009	Srinivasan et al.
2009/0106214 A1	4/2009	Jain et al.
2009/0106215 A1	4/2009	Jain et al.
2009/0106218 A1	4/2009	Srinivasan et al.
2009/0106321 A1	4/2009	Das et al.
2009/0106440 A1	4/2009	Srinivasan et al.
2009/0112802 A1	4/2009	Srinivasan et al.
2009/0112803 A1	4/2009	Srinivasan et al.
2009/0112853 A1	4/2009	Nishizawa et al.
2009/0125550 A1	5/2009	Barga et al.
2009/0133041 A1	5/2009	Rahman et al.
2009/0144696 A1	6/2009	Andersen
2009/0172014 A1*	7/2009	Huetter .... 707/103 R
2009/0182779 A1	7/2009	Johnson
2009/0187584 A1	7/2009	Johnson et al.
2009/0216747 A1	8/2009	Li et al.
2009/0216860 A1	8/2009	Li et al.
2009/0222730 A1	9/2009	Wixson et al.
2009/0228431 A1	9/2009	Dunagan et al.
2009/0228434 A1	9/2009	Krishnamurthy et al.

(56)   References Cited

U.S. PATENT DOCUMENTS

2009/0245236 A1   10/2009 Scott et al.
2009/0248749 A1   10/2009 Gu et al.
2009/0254522 A1   10/2009 Chaudhuri et al.
2009/0257314 A1*   10/2009 Davis et al. 367/125
2009/0265324 A1   10/2009 Mordvinov et al.
2009/0271529 A1   10/2009 Kashiyama et al.
2009/0282021 A1   11/2009 Bennet et al.
2009/0293046 A1   11/2009 Cheriton
2009/0300093 A1   12/2009 Griffiths et al.
2009/0300181 A1   12/2009 Marques
2009/0300580 A1   12/2009 Heyhoe et al.
2009/0300615 A1   12/2009 Andrade et al.
2009/0313198 A1   12/2009 Kudo et al.
2009/0319501 A1   12/2009 Goldstein et al.
2009/0327102 A1   12/2009 Maniar et al.
2010/0017379 A1   1/2010 Naibo et al.
2010/0017380 A1   1/2010 Naibo et al.
2010/0023498 A1   1/2010 Dettinger et al.
2010/0036803 A1   2/2010 Vemuri et al.
2010/0036831 A1   2/2010 Vemuri
2010/0049710 A1   2/2010 Young, Jr. et al.
2010/0057663 A1   3/2010 Srinivasan et al.
2010/0057727 A1   3/2010 Srinivasan et al.
2010/0057735 A1   3/2010 Srinivasan et al.
2010/0057736 A1   3/2010 Srinivasan et al.
2010/0057737 A1   3/2010 Srinivasan et al.
2010/0094838 A1   4/2010 Kozak
2010/0106946 A1   4/2010 Imaki et al.
2010/0125574 A1   5/2010 Navas
2010/0125584 A1   5/2010 Navas
2010/0138405 A1   6/2010 Mihaila
2010/0161589 A1   6/2010 Nica et al.
2010/0223305 A1   9/2010 Park et al.
2010/0223437 A1   9/2010 Park et al.
2010/0223606 A1   9/2010 Park et al.
2010/0293135 A1   11/2010 Candea et al.
2010/0312756 A1   12/2010 Zhang et al.
2010/0318652 A1   12/2010 Samba
2010/0332401 A1   12/2010 Prahlad et al.
2011/0004621 A1   1/2011 Kelley et al.
2011/0016160 A1   1/2011 Zhang et al.
2011/0022618 A1   1/2011 Thatte et al.
2011/0023055 A1   1/2011 Thatte et al.
2011/0029484 A1   2/2011 Park et al.
2011/0029485 A1   2/2011 Park et al.
2011/0040746 A1   2/2011 Handa et al.
2011/0055192 A1   3/2011 Tang et al.
2011/0055197 A1   3/2011 Chavan
2011/0093162 A1   4/2011 Nielsen et al.
2011/0105857 A1   5/2011 Zhang et al.
2011/0161321 A1   6/2011 De Castro Alves et al.
2011/0161328 A1   6/2011 Park et al.
2011/0161352 A1   6/2011 De Castro Alves et al.
2011/0161356 A1   6/2011 De Castro Alves et al.
2011/0161397 A1   6/2011 Bekiares et al.
2011/0173231 A1   7/2011 Drissi et al.
2011/0173235 A1   7/2011 Aman et al.
2011/0196839 A1   8/2011 Smith et al.
2011/0196891 A1   8/2011 De Castro Alves et al.
2011/0270879 A1   11/2011 Srinivasan et al.
2011/0282812 A1   11/2011 Chandramouli et al.
2011/0302164 A1   12/2011 Krishnamurthy et al.
2011/0313844 A1   12/2011 Chandramouli et al.
2011/0314019 A1   12/2011 Jimenez Peris et al.
2011/0321057 A1*   12/2011 Mejdrich et al. 718/105
2012/0041934 A1   2/2012 Srinivasan et al.
2012/0130963 A1   5/2012 Luo et al.
2012/0166417 A1   6/2012 Chandramouli et al.
2012/0166421 A1   6/2012 Cammert et al.
2012/0166469 A1   6/2012 Cammert et al.
2012/0191697 A1   7/2012 Sherman et al.
2012/0233107 A1   9/2012 Roesch et al.
2012/0259910 A1   10/2012 Andrade et al.
2012/0278473 A1   11/2012 Griffiths
2012/0290715 A1   11/2012 Dinger et al.

2012/0324453 A1   12/2012 Chandramouli et al.
2013/0031567 A1   1/2013 Nano et al.
2013/0046725 A1   2/2013 Cammert et al.
2013/0117317 A1   5/2013 Wolf
2013/0144866 A1   6/2013 Jerzak et al.
2013/0191370 A1   7/2013 Chen et al.
2013/0332240 A1   12/2013 Patri et al.
2014/0095444 A1   4/2014 Deshmukh et al.
2014/0095445 A1   4/2014 Deshmukh et al.
2014/0095446 A1   4/2014 Deshmukh et al.
2014/0095447 A1   4/2014 Deshmukh et al.
2014/0095462 A1   4/2014 Park et al.
2014/0095471 A1   4/2014 Deshmukh et al.
2014/0095473 A1   4/2014 Srinivasan et al.
2014/0095483 A1   4/2014 Toillion et al.
2014/0095525 A1   4/2014 Hsiao et al.
2014/0095529 A1   4/2014 Deshmukh et al.
2014/0095533 A1   4/2014 Shukla et al.
2014/0095535 A1   4/2014 Deshmukh et al.
2014/0095537 A1   4/2014 Park et al.
2014/0095540 A1   4/2014 Hsiao et al.
2014/0095541 A1   4/2014 Herwadkar et al.
2014/0095543 A1   4/2014 Hsiao et al.
2014/0136514 A1   5/2014 Jain et al.
2014/0156683 A1   6/2014 de Castro Alves
2014/0172914 A1   6/2014 Elnikety et al.
2014/0201355 A1   7/2014 Bishnoi et al.
2014/0236983 A1   8/2014 Alves et al.
2014/0237289 A1   8/2014 de Castro Alves et al.
2014/0379712 A1   12/2014 Lafuente Alvarez
2015/0156241 A1   6/2015 Shukla et al.
2015/0161214 A1   6/2015 Kali et al.

FOREIGN PATENT DOCUMENTS

WO   WO 00/49533 A2   8/2000
WO   0118712   3/2001
WO   WO 01/59602 A2   8/2001
WO   WO 01/65418 A2   9/2001
WO   WO 03/030031 A2   4/2003
WO   2007122347   11/2007
WO   2012050582   4/2012

OTHER PUBLICATIONS

Martin et al "Finding application errors and security flaws using PQL: a program query language," Proceedings of the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications 40:1-19 (Oct. 2005).
Office Action for U.S. Appl. No. 12/534,384 (Feb. 28, 2012).
Office Action for U.S. Appl. No. 12/506,905 (Mar. 26, 2012).
Office Action for U.S. Appl. No. 12/548,209 (Apr. 16, 2012).
Notice of Allowance for U.S. Appl. No. 13/184,528 (Mar. 1, 2012).
Office Action for U.S. Appl. No. 12/548,187 (Jun. 20, 2012).
Notice of Allowance for U.S. Appl. No. 12/395,871 (May 4, 2012).
Office Action for U.S. Appl. No. 12/548,222 (Jun. 20, 2012).
Office Action for U.S. Appl. No. 12/534,398 (Jun. 5, 2012).
Office Action for U.S. Appl. No. 12/548,281 (Jun. 20, 2012).
Office Action for U.S. Appl. No. 12/913,636 (Jun. 7, 2012).
Esper Reference Documentation, Copyright 2009, ver. 3.1.0, 293 pages.
International Search Report dated Jul. 16, 2012 for PCT/US2012/034970.
Final Office Action for U.S. Appl. No. 12/548,290 dated Jul. 30, 2012.
Office Action for U.S. Appl. No. 13/193,377 dated Aug. 23, 2012.
Office Action for U.S. Appl. No. 11/977,437 dated Aug. 3, 2012.
Final Office Action for U.S. Appl. No. 11/601,415 dated Jul. 2, 2012.
Notice of Allowance for U.S. Appl. No. 12/506,891 dated Jul. 25, 2012.
Final Office Action for U.S. Appl. No. 12/506,905 dated Aug. 9, 2012.
Abadi, D. et al., "Aurora: A Data Stream Management System," International Conference on Management of Data, Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, ACM Press, 2003, 4 pages.

(56)   **References Cited**

OTHER PUBLICATIONS

Aho, A. et al., "Efficient String Matching: An Aid to Bibliographic Search," Communications of the ACM, Jun. 1975, vol. 18, No. 6, pp. 333-340, Copyright 1975, Association for Computing Machinery, Inc.

Arasu, "CQL: A language for Continuous Queries over Streams and Relations," Lectures Notes in Computer Science, 2004, vol. 2921/2004, pp. 1-19.

Arasu, A., et al., "The CQL Continuous Query Language: Semantic Foundations and Query Execution," Stanford University, The VLDB Journal—The International Journal on Very Large Data Bases, vol. 15, Issue 2, Springer-Verlag New York, Inc., Jun. 2006, pp. 1-32.

Arasu, et al., "An Abstract Semantics and Concrete Language for Continuous Queries over Streams and Relations," 9th International Workshop on Database programming languages, Sep. 2003, 12 pages.

Arasu, et al., "STREAM: The Stanford Data Stream Management System," Department of Computer Science, Stanford University, 2004, p. 21.

Avnur, et al., "Eddies: Continuously Adaptive Query Processing," In Proceedings of the 2000 ACM SIGMOD International Conference on Data, Dallas TX, May 2000, 12 pages.

Avnur, et al., "Eddies: Continuously Adaptive Query Processing," slide show, believed to be prior to Oct. 17, 2007, 4 pages.

Babu, et al., "Continuous Queries over Data Streams," SIGMOD Record, Sep. 2001, vol. 30, No. 3, pp. 109-120.

Bai, Y., et al., "A Data Stream Language and System Designed for Power and Extensibility," Conference on Information and Knowledge Management, Proceedings of the 15th ACM International Conference on Information and Knowledge Management, Arlington, Virginia, Nov. 5-11, 2006, 10 pages, ACM Press, Copyright 2006.

Bose, S. et al., "A Query Algebra for Fragmented XML Stream Data", 9th International Conference on Data Base Programming Languages (DBPL), Sep. 6-8, 2003, Postdam, Germany, at URL: http://lambda,uta.edu/dbp103.pdf, 11 pages.

Buza, "Extension of CQL over Dynamic Databases," Journal of Universal Computer Science, 2006, vol. 12, No. 9, pp. 1165-1176.

Carpenter, "User Defined Functions," Oct. 12, 2000, at URL: http://www.sqlteam.com/itemprint.asp?ItemID=979, 4 pages.

Chan, et al., "Efficient Filtering of XML documents with Xpath expressions," VLDB Journal, 2002, pp. 354-379.

Chandrasekaran, et al., "TelegraphCQ: Continuous Dataflow Processing for an Uncertain World," Proceedings of CIDR 2003, p. 12.

Chen, J., et al., "NiagaraCQ: A Scalable Continuous Query System for Internet Databases," Proceedings of the 2000 SIGMOD International Conference on Management of Data, May 2000, pp. 379-390.

Colyer, et al., "Spring Dynamic Modules Reference Guide," Copyright 2006-2008, ver. 1.0.3, 73 pages.

Colyer, et al., "Spring Dynamic Modules Reference Guide," Copyright 2006-2008, ver. 1.1.3, 96 pages.

"Complex Event Processing in the Real World," an Oracle White Paper, Sep. 2007, 13 pages; McReynolds.

Conway, N., "An Introduction to Data Stream Query Processing," Truviso, Inc., May 24, 2007, 71 pages, downloaded from: http://neilconway.org/talks/stream_intro.pdf.

"Coral8 Complex Event Processing Technology Overview," Coral8, Inc., Make it Continuous, pp. 1-8, Copyright 2007 Coral8, Inc.

"Creating WebLogic Domains Using the Configuration Wizard," BEA Products, Dec. 2007, ver. 10.0, 78 pages.

"Creating Weblogic Event Server Applications," BEA WebLogic Event Server, Jul. 2007, ver. 2.0, 90 pages.

Demers, A. et al., "Towards Expressive Publish/Subscribe Systems," in Proceedings of the 10th International Conference on Extending Database Technology (EDBT 2006), Munich, Germany, Mar. 2006, pp. 1-18.

DeMichiel, et al., "JSR 220: Enterprise JavaBeans™, EJB 3.0 Simplified API," EJB 3.0 Expert Group, Sun Microsystems, May 2, 2006, ver. 3.0, 59 pages.

"Dependency Injection," Wikipedia, Dec. 30, 2008, printed on Apr. 29, 2011, at URL: http://en.wikipedia.org/w/index.php?title=Dependency_injection&oldid=260831402, pp. 1-7.

"Deploying Applications to WebLogic Server," BEA WebLogic Server, Mar. 30, 2007, ver. 10.0, 164 pages.

Deshpande, et al., "Adaptive Query Processing," slide show believed to be prior to Oct. 17, 2007, 27 pages.

"Developing Applications with Weblogic Server," BEA WebLogic Server, Mar. 30, 2007, ver. 10.0, 254 pages.

Diao, Y. "Query Processing for Large-Scale XML Message Brokering", 2005, University of California Berkeley, 226 pages.

Diao, Y. et al. "Query Processing for High-Volume XML Message Brokering", Proceedings of the 29th VLDB Conference, Berlin, Germany, 2003, 12 pages.

Dindar, N., et al., "Event Processing Support for Cross-Reality Environments," Pervasive Computing, IEEE CS, Jul.-Sep. 2009, pp. 2-9, Copyright 2009 IEEE.

"EPL Reference," BEA WebLogic Event Server, Jul. 2007, ver. 2.0, 82 pages.

Esper Reference Documentation, Copyright 2007, ver. 1.12.0, 158 pages.

Esper Reference Documentation, Copyright 2008, ver. 2.0.0, 202 pages.

"Fast Track Deployment and Administrator Guide for BEA WebLogic Server," BEA WebLogic Server 10.0 Documentation, printed on May 10, 2010, at URL: http://download.oracle.com/docs/cd/E13222_01/wls/docs100/quickstart/quick_start.html, 1 page.

Fernandez, Mary et al., "Build your own XQuery processor", slide show, at URL: http://www.galaxquery.org/slides/edbt-summerschool2004.pdf, 2004, 116 pages.

Fernandez, Mary et al., Implementing XQuery 1.0: The Galax Experience: Proceedings of the 29th VLDB Conference, Berlin, Germany, 2003, 4 pages.

Florescu, Daniela et al., "The BEA/XQRL Streaming XQuery Processor", Proceedings of the 29th VLDB Conference, 2003, Berlin, Germany, 12 pages.

"Getting Started with WebLogic Event Server," BEA WebLogic Event Server, Jul. 2007, ver. 2.0, 66 pages.

Gilani, A. Design and implementation of stream operators, query instantiator and stream buffer manager, Dec. 2003, 137 pages.

Golab, "Sliding Window Query Processing Over Data Streams," University of Waterloo, Waterloo, Ont. Canada, Aug. 2006, 182 pages.

Golab, L., et al., "Issues in Data Stream Management," ACM SIGMOD Record, vol. 32, Issue 2, Jun. 2003, ACM Press, pp. 5-14.

Gosling, et al., "The Java Language Specification," Book, copyright 1996-2005, 3rd edition, 684 pages, Sun Microsystems USA. (due to size, reference will be uploaded in two parts).

Hopcroft, J. E., "Introduction to Automata Theory, Languages, and Computation," Second Edition, Addison-Wesley, Copyright 2001, 1-521 pages. (due to size, reference will be uploaded in two parts).

"Installing Weblogic Real Time," BEA WebLogic Real Time, Jul. 2007, ver. 2.0, 64 pages.

"Introduction to BEA WebLogic Server and Bea WebLogic Express," BEA WebLogic Server, Mar. 2007, ver. 10.0, 34 pages.

"Introduction to WebLogic Real Time," BEA WebLogic Real Time, Jul. 2007, ver. 2.0, 20 pages.

"Jboss Enterprise Application Platform 4.3 Getting Started Guide CP03, for Use with Jboss Enterprise Application Platform 4.3 Cumulative Patch 3," Jboss a division of Red Hat, Red Hat Documentation Group, Publication date Sep. 2007, Copyright 2008, 68 pages, Red Hat, Inc.

Jin, C. et al. "ARGUS: Efficient Scalable Continuous Query Optimization for Large-Volume Data Streams" 10th International Database Engineering and Applications Symposium (IDEAS'06), 2006, 7 pages.

Knuth, D. E., et al., "Fast Pattern Matching in Strings," SIAM J. COMPUT., vol. 6, No. 2, Jun. 1977, pp. 323-350.

Lakshmanan, et al., "On efficient matching of streaming XML documents and queries," 2002, 18 pages.

Lindholm, et al., "Java Virtual Machine Specification, 2nd Edition", Prentice Hall, Apr. 1999, 484 pages. (due to size, reference will be uploaded in two parts).

(56)   **References Cited**

OTHER PUBLICATIONS

Liu, et al., "Efficient XSLT Processing in Relational Database System," Proceeding of the 32nd. International Conference on Very Large Data Bases (VLDB), Sep. 2006, 1106-1116, 11 pages.
Luckham, D., "What's the Difference Between ESP and CEP?" Complex Event Processing, downloaded Apr. 29, 2011, 5 pages, at URL: http://complexevents.com/?p=103.
Madden, et al., "Continuously Adaptive Continuous Queries (CACQ) over Streams," SIGMOD 2002, Jun. 4-6, 2002, 12 pages.
"Managing Server Startup and Shutdown," BEA WebLogic Server, Mar. 30, 2007, ver. 10.0, 134 pages.
"Matching Behavior," .NET Framework Developers Guide, pp. 1-2, Copyright 2008 Microsoft Corporation, downloaded Jul. 1, 2008 from URL: http://msdn.microsoft.com/en-us/library/0yzc2yb0(printer).aspx.
Motwani, et al., "Models and Issues in Data Streams," Proceedings of the 21st ACM SIGMOD-SIGACT-SIDART symposium on Principles of database systems, 2002, 30 pages.
Motwani, et al., "Query Processing Resource Management, and Approximation in a Data Stream Management System," Proceedings of CIDR 2003, Jan. 2003, 12 pages.
Munagala, et al., "Optimization of Continuous Queries with Shared Expensive Filters," Proceedings of the 26th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, believed to be prior to Oct. 17, 2007, p. 14.
"New Project Proposal for Row Pattern Recognition—Amendment to SQL with Application to Streaming Data Queries," H2-2008-027, H2 Teleconference Meeting, Jan. 9, 2008, pp. 1-6.
Novick, "Creating a User Defined Aggregate with SQL Server 2005," at URL: http://novicksoftware.com/Articles/sql-2005-product-user-defined-aggregate.html, 2005, 6 pages.
Oracle Database, SQL Language Reference, 11g Release 1 (11.1), B28286-02, Sep. 2007, 1496 pages.
Oracle Application Server 10g, Release 2 and 3, New Features Overview, An Oracle White Paper, Oct. 2005, 48 pages.
Oracle Application Server, Administrators Guide, 10g Release 3 (10.1.3.2.0), B32196-01, Jan. 2007, 376 pages, Oracle.
Oracle Application Server, Enterprise Deployment Guide, 10g Release 3 (10.1.3.2.0), B32125-02, Apr. 2007, 120 pages, Oracle.
Oracle Application Server, High Availability Guide, 10g Release 3 (10.1.3.2.0), B32201-01, Jan. 2007, 314 pages, Oracle.
"Oracle CEP Getting Started," Release 11gR1 (11.1.1) E14476-01, May 2009, 172 pages.
Oracle Database Data Cartridge Developers Guide, B28425-03, 11g Release 1 (11.1), Oracle, Mar. 2008, 372 pages.
Oracle Database, SQL Reference, 10g Release 1 (10.1), Part No. B10759-01, Dec. 2003, pp. 7-1 to 7-17; 7-287 to 7-290; 14-61 to 14-74. (due to size, reference will be uploaded in three parts).
"OSGI Service Platform Core Specification, The OSGI Alliance," Apr. 2007, ver. 4.1, release 4, 288 pages, OSGI Alliance.
Peng, et al., "Xpath Queries on Streaming Data," 2003, pp. 1-12, ACM Press.
Peterson, "Petri Net Theory and the Modeling of Systems", Prentice Hall, 1981, 301 pages.
PostgresSQL: Documentation: Manuals: PostgresSQL 8.2: CREATE AGGREGATE, believed to be prior to Apr. 21, 2007, 4 pages.
PostgresSQL: Documentation: Manuals: PostgresSQL 8.2: User-Defined Aggregates, believed to be prior to Apr. 21, 2007, 4 pages.
"Release Notes," BEA WebLogic Event Server, Jul. 2007, ver. 2.0, 8 pages.
Sadri, R., et al., "Expressing and Optimizing Sequence Queries in Database Systems," ACM Transactions on Database Systems, vol. 29, No. 2, Jun. 2004, pp. 282-318, ACM Press, Copyright 2004.
Sadtler, et al., "WebSphere Application Server Installation Problem Determination," Copyright 2007, pp. 1-48, IBM Corp.
Sharaf, et al., Efficient Scheduling of Heterogeneous Continuous Queries, VLDB '06, Sep. 12-15, 2006, pp. 511-522.
Spring Dynamic Modules for OSGi Service Platforms product documentation, SpringSource, Jan. 2008, 71 pages; Colyer et al.

"Stanford Stream Data Manager," at URL: http://infolab.stanford.edu/stream/, last modified Jan. 5, 2006, pp. 1-9.
Stolze, "User-defined Aggregate Functions in DB2 Universal Database," at URL: http://www.128.ibm.com/developerworks/db2/library/tacharticle/0309stolze/0309stolze.html, Sep. 11, 2003, 9 pages.
Stream Query Repository: Online Auctions (CQL Queries), at URL: http://www-db.stanford.edu/strem/sqr/cql/onauc.html, Dec. 2, 2002, 4 pages.
Stream Query Repository: Online Auctions, at URL: http://www-db.stanford.edu/stream/sqr/onauc.html#queryspecsend, Dec. 2, 2002, 2 pages.
"Stream: The Stanford Stream Data Manager," IEEE Data Engineering Bulletin, Mar. 2003, pp. 1-8.
"StreamBase New and Noteworthy," StreamBase, dated Jan. 12, 2010, 878 pages.
Terry, et al., "Continuous queries over append-only database," Proceedings of 1992 ACM SIGMOD, pp. 321-330.
"Understanding Domain Configuration," BEA WebLogic Server, Mar. 30, 2007, ver. 10.0, 38 pages.
Vajjhala, et al, "The Java™ Architecture for XML Binding (JAXB) 2.0," Sun Microsystem, Inc., Final Release Apr. 19, 2006, 384 pages.
W3C, "XML Path Language (Xpath)," W3C Recommendation, Nov. 16, 1999, ver. 1.0, at URL: http://www.w3.org/TR/xpath, 37 pages.
"WebLogic Event Server Administration and Configuration Guide," BEA WebLogic Event Server, Jul. 2007, ver. 2.0, 108 pages.
"WebLogic Event Server Reference," BEA WebLogic Event Server, Jul. 2007, ver. 2.0, 52 pages.
"Weblogic Server Performance and Tuning," BEA WebLogic Server, Mar. 30, 2007, ver. 10.0, 180 pages.
"WebSphere Application Server V6.1 Problem Determination: IBM Redpaper Collection," WebSphere Software, IBM/Redbooks, Dec. 2007, 634 pages.
White, S., et al., "WebLogic Event Server: A Lightweight, Modular Application Server for Event Processing," 2nd International Conference on Distributed Event-Based Systems, Jul. 2-4, 2008, Rome, Italy, 8 pages, ACM Press, Copyright 2004.
Widom, et al., "CQL: A Language for Continuous Queries over Streams and Relations," believed to be prior to Oct. 17, 2007, 62 pages.
Widom, et al., "The Stanford Data Stream Management System," PowerPoint Presentation, believed to be prior to Oct. 17, 2007, 110 pages.
Zemke,"XML Query," Mar. 14, 2004, 29 pages.
Non-Final Office Action for U.S. Appl. No. 12/548,187, mailed on Sep. 27, 2011, 19 pages.
Non-Final Office Action for U.S. Appl. No. 12/396,008, mailed on Jun. 8, 2011, 10 pages.
Non-Final Office Action for U.S. Appl. No. 12/395,871, mailed on May 27, 2011, 7 pages.
Final Office Action for U.S. Appl. No. 12/395,871, mailed on Oct. 19, 2011, 33 pages.
Non-Final Office Action for U.S. Appl. No. 12/548,222, mailed on Oct. 17, 2011, 27 pages.
Non-Final Office Action for U.S. Appl. No. 12/548,281, mailed on Oct. 3, 2011, 37 pages.
Non-Final Office Action for U.S. Appl. No. 12/548,290, mailed on Oct. 3, 2011, 34 pages.
Non-Final Office Action for U.S. Appl. No. 11/874,202, mailed on Dec. 3, 2009, 20 pages.
Final Office Action for U.S. Appl. No. 11/874,202, mailed on Jun. 8, 2010, 18 pages.
Notice of Allowance for U.S. Appl. No. 11/874,202, mailed on Dec. 22, 2010, 23 pages.
Notice of Allowance for U.S. Appl. No. 11/874,202, mailed on Mar. 31, 2011, 12 pages.
Notice of Allowance for U.S. Appl. No. 11/874,850, mailed on Nov. 24, 2009, 17 pages.
Supplemental Notice of Allowance for U.S. Appl. No. 11/874,850, mailed on Dec. 11, 2009, 5 pages.
Supplemental Notice of Allowance for U.S. Appl. No. 11/874,850, mailed on Jan. 27, 2010, 11 pages.

(56) **References Cited**

OTHER PUBLICATIONS

Non-Final Office Action for U.S. Appl. No. 11/874,896, mailed on Dec. 8, 2009, 19 pages.
Final Office Action for U.S. Appl. No. 11/874,896, mailed on Jul. 23, 2010, 28 pages.
Non-Final Office Action for U.S. Appl. No. 11/874, 896, mailed on Nov. 22, 2010, 29 pages.
Notice of Allowance for U.S. Appl. No. 11/874,896, mailed on Jun. 23, 2011, 30 pages.
Non-Final Office Action for U.S. Appl. No. 11/977,439, mailed on Apr. 13, 2010, 11 pages.
Notice of Allowance for U.S. Appl. No. 11/977,439, mailed on Aug. 18, 2010, 11 pages.
Supplemental Notice of Allowance for U.S. Appl. No. 11/977,439, mailed on Sep. 28, 2010, 6 pages.
Notice of Allowance for U.S. Appl. No. 11/977,439, mailed on Nov. 24, 2010, 14 pages.
Notice of Allowance for U.S. Appl. No. 11/977,439, mailed on Mar. 16, 2011, 15 pages.
Non-Final Office Action for U.S. Appl. No. 11/977,437, mailed on Oct. 13, 2009, 12 pages.
Final Office Action for U.S. Appl. No. 11/977,437, mailed on Apr. 8, 2010, 20 pages.
Notice of Allowance for U.S. Appl. No. 11/977,440, mailed on Oct. 7, 2009, 9 pages.
Office Action for U.S. Appl. No. 11/874,197, mailed on Nov. 10, 2009, 17 pages.
Final Office Action for U.S. Appl. No. 11/874,197, mailed on Jun. 29, 2010, 19 pages.
Non-Final Office Action for U.S. Appl. No. 11/874,197, mailed on Dec. 22, 2010, 29 pages.
Final Office Action for U.S. Appl. No. 11/874,197, mailed on Aug. 12, 2011, 26 pages.
Non-Final Office Action for U.S. Appl. No. 11/873,407, mailed on Nov. 13, 2009, 10 pages.
Final Office Action for U.S. Appl. No. 11/873,407, mailed on Apr. 26, 2010, 12 pages.
Notice of Allowance for U.S. Appl. No. 11/873,407, mailed on Nov. 10, 2010, 14 pages.
Notice of Allowance for U.S. Appl. No. 11/873,407, mailed on Mar. 7, 2011, 11 pages.
Non-Final Office Action for U.S. Appl. No. 11/601,415, mailed on Sep. 17, 2008, 13 pages.
Final Office Action for U.S. Appl. No. 11/601,415, mailed on May 27, 2009, 30 pages.
Advisory Action for U.S. Appl. No. 11/601,415, mailed on Aug. 18, 2009, 3 pages.
Non-Final Office Action for U.S. Appl. No. 11/601,415, mailed on Nov. 30, 2009, 33 pages.
Final Office Action for U.S. Appl. No. 11/601,415, mailed on Jun. 30, 2010, 45 pages.
Non-Final Office Action for U.S. Appl. No. 11/927,681, mailed on Mar. 24, 2011, 17 pages.
Notice of Allowance for U.S. Appl. No. 11/927,681, mailed on Jul. 1, 2011, 8 pages.
Non-Final Office Action for U.S. Appl. No. 11/927,683, mailed on Mar. 24, 2011, 13 pages.
Final Office Action for U.S. Appl. No. 11/927,683, mailed on Sep. 1, 2011, 18 pages.
Non-Final Office Action for U.S. Appl. No. 10/948,523, mailed on Jan. 22, 2007, 55 pages.
Final Office Action for U.S. Appl. No. 10/948,523, mailed on Jul. 6, 2007, 42 pages.
Non-Final Office Action for U.S. Appl. No. 10/948,523, mailed on Dec. 11, 2007, 52 pages.
Notice of Allowance for U.S. Appl. No. 10/948,523, mailed on Jul. 8, 2008, 38 pages.
Supplemental Notice of Allowance for U.S. Appl. No. 10/948,523, mailed on Aug. 25, 2008, 3 pages.

Notice of Allowance for U.S. Appl. No. 10/948,523, mailed on Dec. 1, 2010, 18 pages.
International Search Report dated Sep. 12, 2012 for PCT/US2012/036353.
Office Action for U.S. Appl. No. 13/244,272 dated Oct. 4, 2012.
Notice of Allowance for U.S. Appl. No. 12/548,209 dated Oct. 24, 2012.
Nah et al. "A Cluster-Based THO-Structured Scalable Approach for Location Information Systems," The Ninth IEEE International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS' 03), pp. 225-233 (Jan. 1, 2003).
Hulton et al. "Mining Time-Changing Data Streams," Proceedings of the Seventh ACM SIGKDD, pp. 10 (Aug. 2001).
Stump et al. (ed.) Proceedings of IJCAR '06 Workshop "PLPV '06: Programming Languages meets Program Verification," pp. 1-113 (Aug. 21, 2006).
Vijayalakshmi et al. "Processing location dependent continuous queries in distributed mobile databases using mobile agents," IET-UK International Conference on Information and Communication Technology in Electrical Sciences (ICTES 2007), pp. 1023-1030 (Dec. 22, 2007).
Wang et al. "Distributed Continuous Range Query Processing on Moving Objects," Proceedings of the 17th international Conference on Database and Expert Systems Applications (DEXA'06), Berlin, DE, pp. 655-665 (Jan. 1, 2006).
Wu et al. "Dynamic Data Management for Location Based Services in Mobile Environments," IEEE Proceedings of the Seventh International Database Engineering and Applications Symposium 2003 Piscataway. NJ. USA., pp. 172-181 (Jul. 16, 2003).
Hao et al. "Achieving high performance web applications by service and database replications at edge servers," proceedings of IPCCC 2009, IEEE 28th International Performance Computing and Communications Conference, pp. 153-160 (Dec. 2009).
International Search Report dated for PCT/US2011/052019 (Nov. 17, 2011).
Office Action for U.S. Appl. No. 12/396,008 (Nov. 16, 2011).
Office Action for U.S. Appl. No. 12/506,891 (Dec. 14, 2011).
Office Action for U.S. Appl. No. 12/534,398 (Nov. 1, 2011).
Office Action for U.S. Appl. No. 11/601,415 (Dec. 9, 2011).
Sansoterra "Empower SOL with Java User-Defined Functions," IT Jungle.com (Oct. 9, 2003).
Ullman et al., "Introduction to JDBC," Stanford University (2005).
Non-Final Office Action for U.S. Appl. No. 12/957,194 dated Dec. 7, 2012.
Non-Final Office Action for U.S. Appl. No. 13/089,556 dated Nov. 6, 2012.
Notice of Allowance for U.S. Appl. No. 12/534,398 dated Nov. 27, 2012.
Notice of Allowance for U.S. Appl. No. 12/506,905 dated Dec. 14, 2012.
Non-Final Office Action for U.S. Appl. No. 12/957,201 dated Dec. 19, 2012.
U.S. Appl. No. 12/534,384, Notice of Allowance mailed on May 7, 2013, 12 pages.
U.S. Appl. No. 12/548,187, Non-Final Office Action mailed on Apr. 9, 2013, 17 pages.
U.S. Appl. No. 12/548,222, Non-Final Office Action mailed on Apr. 10, 2013, 16 pages.
U.S. Appl. No. 12/548,281, Non-Final Office Action mailed on Apr. 13, 2013, 16 pages.
U.S. Appl. No. 12/548,290, Non-Final Office Action mailed on Apr. 15, 2013, 17 pages.
U.S. Appl. No. 12/957,201, Final Office Action mailed on Apr. 25, 2013, 11 pages.
U.S. Appl. No. 13/089,556, Non-Final Office Action mailed on Apr. 10, 2013, 10 pages.
Notice of Allowance for U.S. Appl. No. 11/977,437 dated Mar. 4, 2013. 9 pages.
Final Office Action for U.S. Appl. No. 13/244,272 dated Mar. 28, 2013, 29 pages.
Notice of Allowance for U.S. Appl. No. 12/957,194 dated Mar. 20, 2013. 9 pages.

(56) **References Cited**

OTHER PUBLICATIONS

Kawaguchi et al. "Java Architecture for XML Binding (JAXB) 2.2, " Sun Microsystems, Inc., Dec. 10, 1999, 384 pages.
Final Office Action for U.S. Appl. No. 12/396,464 dated Jan. 16, 2013, 16 pages.
Final Office Action for U.S. Appl. No. 12/913,636 dated Jan. 8, 2013, 20 pages.
Non-Final Office Action for U.S. Appl. No. 12/949,081 dated Jan. 9, 2013, 12 pages.
Final Office Action for U.S. Appl. No. 13/193,377 dated Jan. 17, 2013, 24 pages.
Final Office Action for U.S. Appl. No. 12/534,384 dated Feb. 12, 2013, 13 pages.
Non-Final Office Action for U.S. Appl. No. 13/102,665 dated Feb. 1, 2013, 11 pages.
Oracle™ Fusion Middleware CQL Language Reference, 11g Release 1 (11.1.1.6.3) E12048-10, Aug. 2012, pp. 6-1 to 6-12.
Oracle™ Complex Event Processing CQL Language Reference, 11g Release 1 (11.1.1.4.0) E12048-04, Jan. 2011, pp. 6.1 to 6.12.
Oracle™ Complex Event Processing CQL Language Reference, 11g Release 1 (11.1.1) E12048-03, Apr. 2010, sections 18-4 to 18.4.2.
Pattern Recognition With MATCH_RECOGNIZE, Oracle™ Complex Event Processing CQL Language Reference, 11g Release 1 (11.1.1) E12048-01, May 2009, pp. 15.1 to 15.20.
Supply Chain Event Management: Real-Time Supply Chain Event Management, product information Manhattan Associates, 2009-2012.
U.S. Appl. No. 11/601,415, Non-Final Office Action mailed on Dec. 11, 2013, 58 pages.
U.S. Appl. No. 12/396,464, Non Final Office Action mailed on Dec. 31, 2013, 16 pages.
U.S. Appl. No. 13/089,556, Non-Final Office Action mailed on Jan. 9, 2014, 14 Pages.
Non-Final Office Action for U.S. Appl. No. 12/548,187 dated Feb. 6, 2014, 53 pages.
Agrawal et al. "Efficient pattern matching over event streams," Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pp. 147-160 (Jun. 2008).
Chandramouli et al., High-Performance Dynamic Pattern Matching over Disordered Streams, Proceedings of the VLDB Endowment, vol. 3, Issue 1-2, Sep. 2010, pp. 220-231.
Chapple "Combining Query Results with the Union Command," ask.com Computing Databases, downloaded from: http://databases.about.com/od/sql/a/union.htm (no date, printed on Oct. 14, 2013).
Fantozzi, A Strategic Approach to Supply Chain Event Management, student submission for Masters Degree, Massachusetts Institute of Technology, Jun. 2003.
Komazec et al., Towards Efficient Schema-Enhanced Pattern Matching over RDF Data Streams, Proceedings of the 1st International Workshop on Ordering and Reasoning (OrdRing 2011), Bonn, Germany, Oct. 2011.
Ogrodnek, Custom UDFs and hive, Bizo development blog http://dev.bizo.com, Jun. 23, 2009, 2 pages.
Pradhan, Implementing and Configuring SAP® Event Management, Galileo Press, 2010, pp. 17-21.
Wilson et al., SAP Event Management, an Overview, Q Data USA, Inc., 2009.
Business Process Management (BPM), Datasheet [online]. IBM, [retrieved on Jan. 28, 2013]. Retrieved from the Internet: <URL: http://www-142.ibm.com/software/products/us/en/category/BPM-SOFTWARE>.
What is BPM? , Datasheet [online]. IBM, [retrieved on Jan. 28, 2013]. Retrieved from the Internet: <URL: http://www-01.ibm.com/software/info/bpm/whatis-bpm/>.
U.S. Appl. No. 12/548,281, Final Office Action mailed on Oct. 10, 2013, 21 pages.
U.S. Appl. No. 12/548,290, Notice of Allowance mailed on Sep. 11, 2013, 6 pages.
U.S. Appl. No. 12/949,081, Final Office Action mailed on Aug. 27, 2013, 13 pages.

U.S. Appl. No. 13/089,556, Final Office Action mailed on Aug. 29, 2013, 10 pages.
U.S. Appl. No. 13/177,748, Non-Final Office Action mailed on Aug. 30, 2013, 24 pages.
U.S. Appl. No. 13/193,377, Notice of Allowance mailed on Aug. 30, 2013, 19 pages.
U.S. Appl. No. 12/548,187, Final Office Action mailed on Jun. 10, 2013, 18 pages.
U.S. Appl. No. 12/548,222, Notice of Allowance mailed on Jul. 18, 2013, 12 pages.
U.S. Appl. No. 13/102,665, Final Office Action mailed on Jul. 9, 2013, 17 pages.
Notice of Allowance for U.S. Appl. No. 11/977,437 dated Jul. 10, 2013, 10 pages.
SQL Tutorial-In, Tizag.com, http://web.archive.org/web/20090216215219/http://www.tizag.com/sgiTutorial/sqlin.php, Feb. 16, 2009, pp. 1-3.
U.S. Appl. No. 12/548,281, Non-Final Office Action mailed on Feb. 13, 2014, 16 pages.
U.S. Appl. No. 13/177,748, Final Office Action mailed on Mar. 20, 2014, 23 pages.
International Search Report dated Apr. 3, 2014 for PCT/US2014/010832, 9 pages.
Cadonna et al. "Efficient event pattern matching with match windows," Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 471-479 (Aug. 2012).
Nichols et al. "A faster closure algorithm for pattern matching in partial-order event data," IEEE International Conference on Parallel and Distributed Systems, pp. 1-9 (Dec. 2007).
Babu et al., "Exploiting k-Constraints to Reduce Memory Overhead in Continuous Queries Over Data Streams", ACM Transactions on Database Systems (TODS) vol. 29 Issue 3, Sep. 2004, 36 pages.
Tho et al. "Zero-latency data warehousing for heterogeneous data sources and continuous data streams," 5th International Conference on Information Integrationand Web-based Applications Services (Sep. 2003) 12 pages.
"SQL Subqueries"—Dec. 3, 2011, 2 pages.
"Caching Data with SqiDataSource Control"—Jul. 4, 2011, 3 pages.
"SCD-Slowing Changing Dimensions in a Data Warehouse"—Aug. 7, 2011, one page.
Non-Final Office Action for U.S. Appl. No. 13/838,259 dated Oct. 24, 2014, 21 pages.
Notice of Allowance for U.S. Appl. No. 13/102,665 dated Nov. 24, 2014, 9 pages.
Non-Final Office Action for U.S. Appl. No. 13/827,631 dated Nov. 13, 2014, 10 pages.
Non-Final Office Action for U.S. Appl. No. 13/827,987 dated Nov. 6, 2014, 9 pages.
Non-Final Office Action for U.S. Appl. No. 11/601,415 dated Oct. 6, 2014, 18 pages.
Non-Final Office Action for U.S. Appl. No. 14/077,230 dated Dec. 4, 2014, 30 pages.
Non-Final Office Action for U.S. Appl. No. 13/828,640 dated Dec. 2, 2014, 11 pages.
Non-Final Office Action for U.S. Appl. No. 13/830,428 dated Dec. 5, 2014, 23 pages.
Non-Final Office Action for U.S. Appl. No. 13/830,502 dated Nov. 20, 2014, 25 pages.
Non-Final Office Action for U.S. Appl. No. 13/839,288 dated Dec. 4, 2014, 30 pages.
Call User Defined Functions from Pig, Amazon Elastic MapReduce, Mar. 2009, 2 pages.
Strings in C, retrieved from the internet: <URL: https://web.archive.org/web/20070612231205/http:l/web.cs.swarthmore.edu/-newhall/unixhelp/C_strings.html> [retrieved on May 13, 2014], Swarthmore College, Jun. 12, 2007, 3 pages.
Non-Final Office Action for U.S. Appl. No. 12/913,636 dated Jul. 24, 2014, 21 pages.
U.S. Appl. No. 11/874,197, Notice of Allowance mailed on Jun. 22, 2012, 20 pages.
U.S. Appl. No. 12/396,464, Final Office Action mailed on May 16, 2014, 16 pages.

(56)　　　　　**References Cited**

OTHER PUBLICATIONS

U.S. Appl. No. 12/396,464, Non-Final Office Action mailed on Sep. 7, 2012, 18 pages.
U.S. Appl. No. 12/548,187, Final Office Action mailed on Jun. 4, 2014, 64 pages.
U.S. Appl. No. 13/089,556, Final Office Action mailed on Jun. 13, 2014, 14 pages.
U.S. Appl. No. 13/244,272, Notice of Allowance mailed on Aug. 12, 2013, 12 pages.
International Application No. PCT/US2011/052019, International Preliminary Report on Patentability mailed on Mar. 28, 2013, 6 pages.
International Application No. PCT/US2012/034970, International Preliminary Report on Patentability mailed on Nov. 21, 2013, 7 pages.
Bottom-up parsing, Wikipedia, downloaded from: http://en.wikipedia.org/wiki/Bottom-up_parsing, Sep. 8, 2014, pp. 1-2.
Branch Predication, Wikipedia, downloaded from: http://en.wikipedia.org/wiki/Branch_predication, Sep. 8, 2014, pp. 1-4.
Microsoft Computer Dictionary, 5th Edition, Microsoft Press, Redmond, WA, 2002, pp. 238-239 and 529.
Notice of Allowance for U.S. Appl. No. 13/089,556 dated Oct. 6, 2014, 9 pages.
U.S. Appl. No. 12/396,464, Notice of Allowance mailed on Sep. 3, 2014, 7 pages.
U.S. Appl. No. 12/548,187, Advisory Action mailed on Sep. 26, 2014, 6 pages.
U.S. Appl. No. 12/548,281, Final Office Action mailed on Aug. 13, 2014, 19 pages.
U.S. Appl. No. 12/957,201, Non-Final Office Action mailed on Jul. 30, 2014, 12 pages.
U.S. Appl. No. 13/764,560, Non-Final Office Action mailed on Sep. 12, 2014, 23 pages.
U.S. Appl. No. 13/770,969, Non-Final Office Action mailed on Aug. 7, 2014, 9 pages.
U.S. Appl. No. 14/302,031, Non-Final Office Action mailed on Aug. 27, 2014, 19 pages.
Abadi et al., Aurora: a new model and architecture for data stream management, The VLDB Journal the International Journal on Very Large Data Bases, vol. 12, No. 2, Aug. 1, 2003, pp. 120-139.
Balkesen et al., Scalable Data Partitioning Techniques for Parallel Sliding Window Processing over Data Streams, 8th International Workshop on Data Management for Sensor Networks, Aug. 29, 2011, pp. 1-6.
Chandrasekaran et al., PSoup: a system for streaming queries over streaming data, The VLDB Journal, The International Journal on Very Large Data Bases, vol. 12, No. 2, Aug. 1, 2003, pp. 140-156.
Dewson, Beginning SQL Server 2008 for Developers: From Novice to Professional, A Press, Berkeley, CA 2008, pp. 337-349 and 418-438.
Harish D et al., Identifying robust plans through plan diagram reduction, PVLDB 08, Auckland, New Zealand, Aug. 23-28, pp. 1124-1140.
Krämer, Continuous Queries Over Data Streams—Semantics and Implementation, Fachbereich Mathematik and Informatik der Philipps-Universitat, Marburg, Germany, Retrieved from the Internet:　　URL:http://archiv.ub.uni-marburg.de/dissjz007/0671/pdfjdjk.pdf, Jan. 1, 2007; 313 pages.
International Application No. PCT/US2013/062047, International Search Report and Written Opinion mailed on Jul. 16, 2014, 12 pages.
International Application No. PCT/US2013/062050, International Search Report & Written Opinion mailed on Jul. 2, 2014, 13 pages.
International Application No. PCT/US2013/062052, International Search Report & Written Opinion mailed on Jul. 3, 2014, 12 pages.
International Application No. PCT/US2013/073086, International Search Report and Written Opinion mailed on Mar. 14, 2014.
International Application No. PCT/US2014/017061, International Search Report and Written Opinion mailed on Sep. 9, 2014, 12 pages.

Rao et al., Compiled Query Execution Engine using JVM, ICDE '06, Atlanta, GA, Apr. 3-7, 2006, 12 pages.
Ray et al., Optimizing complex sequence pattern extraction using caching, data engineering workshops (ICDEW)~ 2011 IEEE 27th international conference on IEEE, Apr. 11, 2011, pp. 243-248.
Shah et al., Flux: an adaptive partitioning operator for continuous query systems, Proceedings of the 19th International Conference on Data Engineering, Mar. 5-8, 2003, pp. 25-36.
Stillger et al., LEO—DB2's Learning Optimizer, Proc. of the VLDB, Roma, Italy, Sep. 2001, pp. 19-28.
Cranor et al. "Gigascope: a stream database for network applications," Proceedings of the 2003 ACM SIGMOD international conference on Management of data, pp. 647-651 (Jun. 2003).
International Application No. PCT/US2014/068641, International Search Report and Written Opinion mailed on Feb. 26, 2015, 11 pages.
European Patent Application No. 12783063.6, Extended Search Report mailed Mar. 24, 2015, 6 pages.
U.S. Appl. No. 12/949,081, Non-Final Office Action mailed on Jan. 28, 2015, 20 pages.
U.S. Appl. No. 12/957,201, Notice of Allowance mailed on Jan. 21, 2015, 5 pages.
U.S. Appl. No. 13/177,748, Non-Final Office Action mailed on Feb. 3, 2015, 22 pages.
U.S. Appl. No. 13/770,961, Non-Final Office Action mailed on Feb. 4, 2015, 22 pages.
U.S. Appl. No. 13/770,969, Notice of Allowance mailed on Jan. 22, 2015, 5 pages.
U.S. Appl. No. 13/829,958, Non-Final Office Action mailed on Dec. 11, 2014, 15 pages.
Non-Final Office Action for U.S. Appl. No. 13/830,378 dated Feb. 25, 2015, 23 pages.
Non-Final Office Action for U.S. Appl. No. 13/830,129 dated Feb. 27, 2015, 19 pages.
Non-Final Office Action for U.S. Appl. No. 12/913,636 dated Apr. 1, 2015, 22 pages.
Final Office Action for U.S. Appl. No. 13/827,631 dated Apr. 3, 2015, 11 pages.
Notice of Allowance for U.S. Appl. No. 13/839,288 dated Apr. 3, 2015, 12 pages.
Notice of Allowance for U.S. Appl. No. 14/077,230 dated Apr. 16, 2015, 16 pages.
Final Office Action for U.S. Appl. No. 13/764,560 dated Apr. 15, 2015, 19 pages.
International Application No. PCT/US2014/010832, Written Opinion mailed on Dec. 15, 2014, 5 pages.
International Application No. PCT/US2014/010920, International Search Report and Written Opinion mailed on Dec. 15, 2014, 10 pages.
International Application No. PCT/US2014/017061, Written Opinion mailed on Feb. 3, 2015, 6 pages.
Oracle® Complex Event Processing EPL Language Reference 11g Release 1 (11.1.1.4.0), E14304-02, Jan. 2011, 80 pages.
De Castro Alves, A General Extension System for Event Processing Languages, DEBS '11, New York, NY, USA, Jul. 11-15, 2011, pp. 1-9.
Takenaka et al., A scalable complex event processing framework for combination of SQL-based continuous queries and C/C++ functions, FPL 2012, Oslo, Norway, Aug. 29-31, 2012, pp. 237-242.
Tomàs et al., RoSeS: A Continuous Content-Based Query Engine for RSS Feeds, DEXA 2011, Toulouse, France, Sep. 2, 2011, pp. 203-218.
Notice of Allowance for U.S. Appl. No. 12/548,187 dated Aug. 17, 2015, 18 pages.
Non-Final Office Action for U.S. Appl. No. 14/037,072 dated Jul. 9, 2015, 12 pages.
Non-Final Office Action for U.S. Appl. No. 13/830,502 dated Jun. 30, 2015, 25 pages.
Non-Final Office Action for U.S. Appl. No. 14/036,659 dated Aug. 13, 2015, 33 pages.
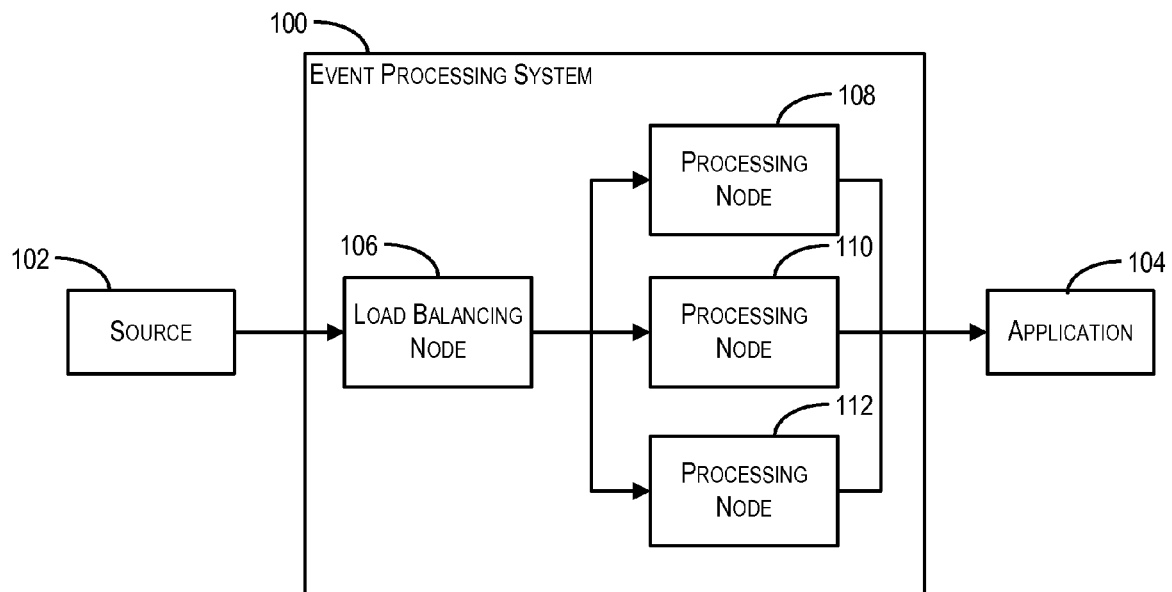Non-Final Office Action for U.S. Appl. No. 13/830,759 dated Aug. 7, 2015, 23 pages.
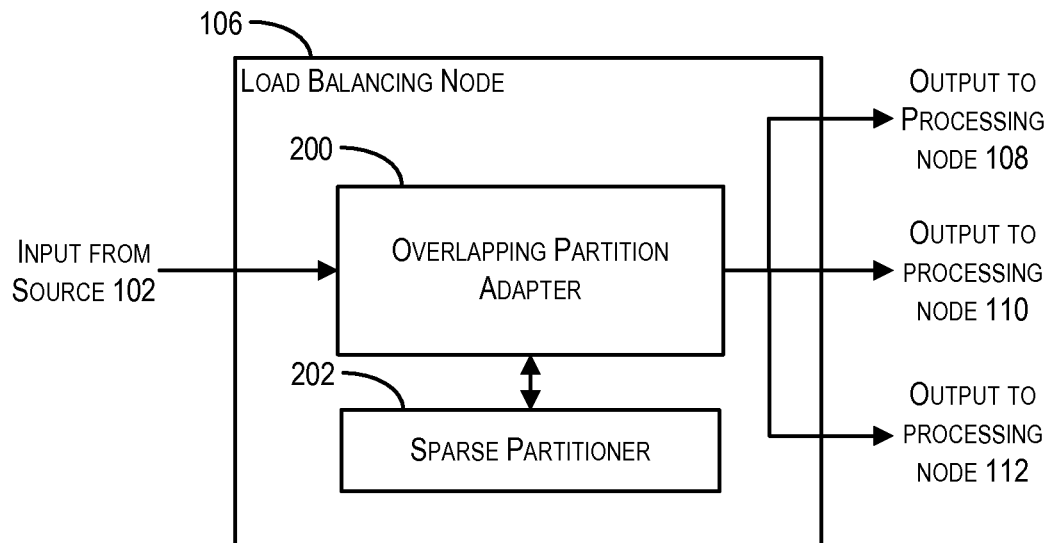
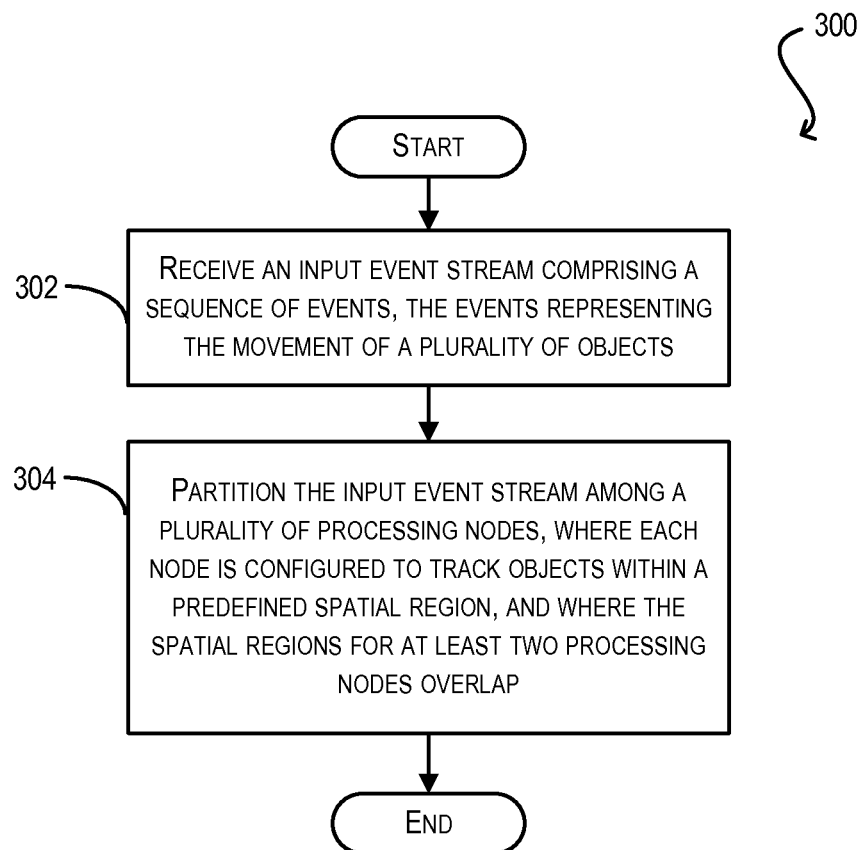(56) **References Cited**

OTHER PUBLICATIONS

Final Office Action for U.S. Appl. No. 14/302,031 dated Apr. 22, 2015, 23 pages.

Non-Final Office Action for U.S. Appl. No. 14/692,674 dated Jun. 5, 2015, 22 pages.

Non-Final Office Action for U.S. Appl. No. 14/037,171 dated Jun. 3, 2015, 15 pages.

Non-Final Office Action for U.S. Appl. No. 14/830,735 dated May 26, 2015, 19 pages.

Final Office Action for U.S. Appl. No. 13/830,428 dated Jun. 4, 2015, 21 pages.

Non-Final Office Action for U.S. Appl. No. 14/838,259 dated Jun. 9, 2015, 37 pages.

Final Office Action for U.S. Appl. No. 14/906,162 dated Jun. 10, 2015, 10 pages.

Non-Final Office Action for U.S. Appl. No. 14/037,153 dated Jun. 19, 2015, 23 pages.

Final Office Action for U.S. Appl. No. 13/829,958 dated Jun. 19, 2015, 17 pages.

Final Office Action for U.S. Appl. No. 13/827,987 dated Jun. 19, 2015, 10 pages.

Final Office Action for U.S. Appl. No. 13/828,640 dated Jun. 17, 2015, 11 pages.

U.S. Appl. No. 13/906,162, Non-Final Office Action mailed on Dec. 29, 2014, 10 pages.
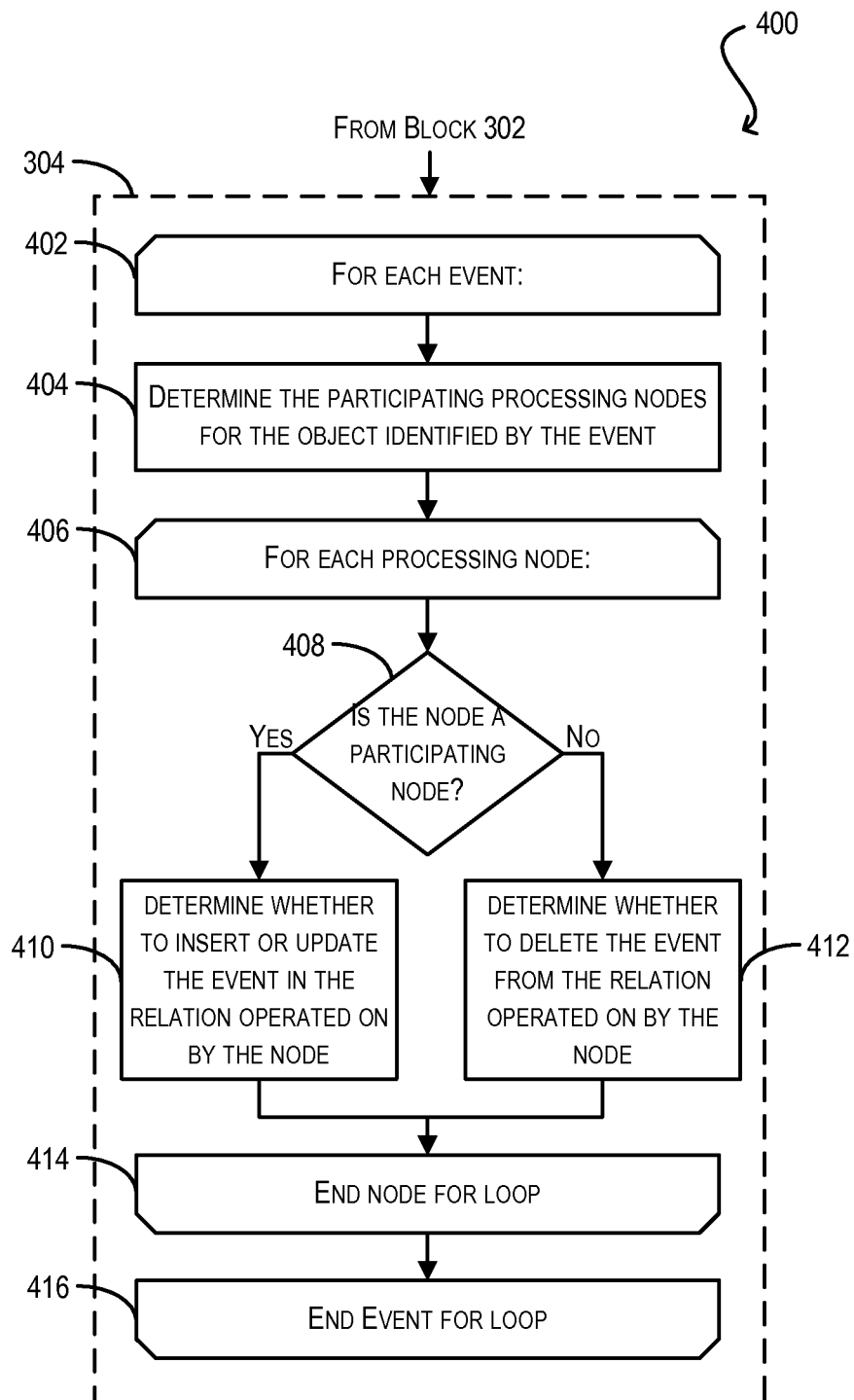
International Application No. PCT/US2014/039771, International Search Report and Written Opinion mailed on Apr. 29, 2015 6 pages.

International Application No. PCT/US2015/016346, International Search Report and Written Opinion mailed on May 4, 2015, 9 pages.

International Preliminary Report on Patentability dated Apr. 9, 2015 for PCT/US2013/062047, 10 pages.

International Preliminary Report on Patentability dated Apr. 9, 2015 for PCT/US2013/062052, 18 pages.

International Preliminary Report on Patentability dated May 28, 2015 for PCT/US2014/017061, 31 pages.

International Preliminary Report on Patentability dated Jun. 18, 2015 for PCT/US2013/073086, 7 pages.

International Preliminary Report on Patentability dated Jul. 29, 2015 for PCT/US2014/010920, 30 pages.

International Preliminary Report on Patentability dated Jul. 29, 2015 for PCT/US2014/039771, 24 pages.

Japan Patent Office office actions JPO patent application JP2013-529376 (Aug. 18, 2015).

Final Office Action for U.S. Appl. No. 13/177,748 dated Aug. 21, 2015, 24 pages.

Non-Final Office Action for U.S. Appl. No. 14/036,500 dated Aug. 14, 2015, 26 pages.

Notice of Allowance for U.S. Appl. No. 13/830,129 dated Sep. 22, 2015, 9 pages.

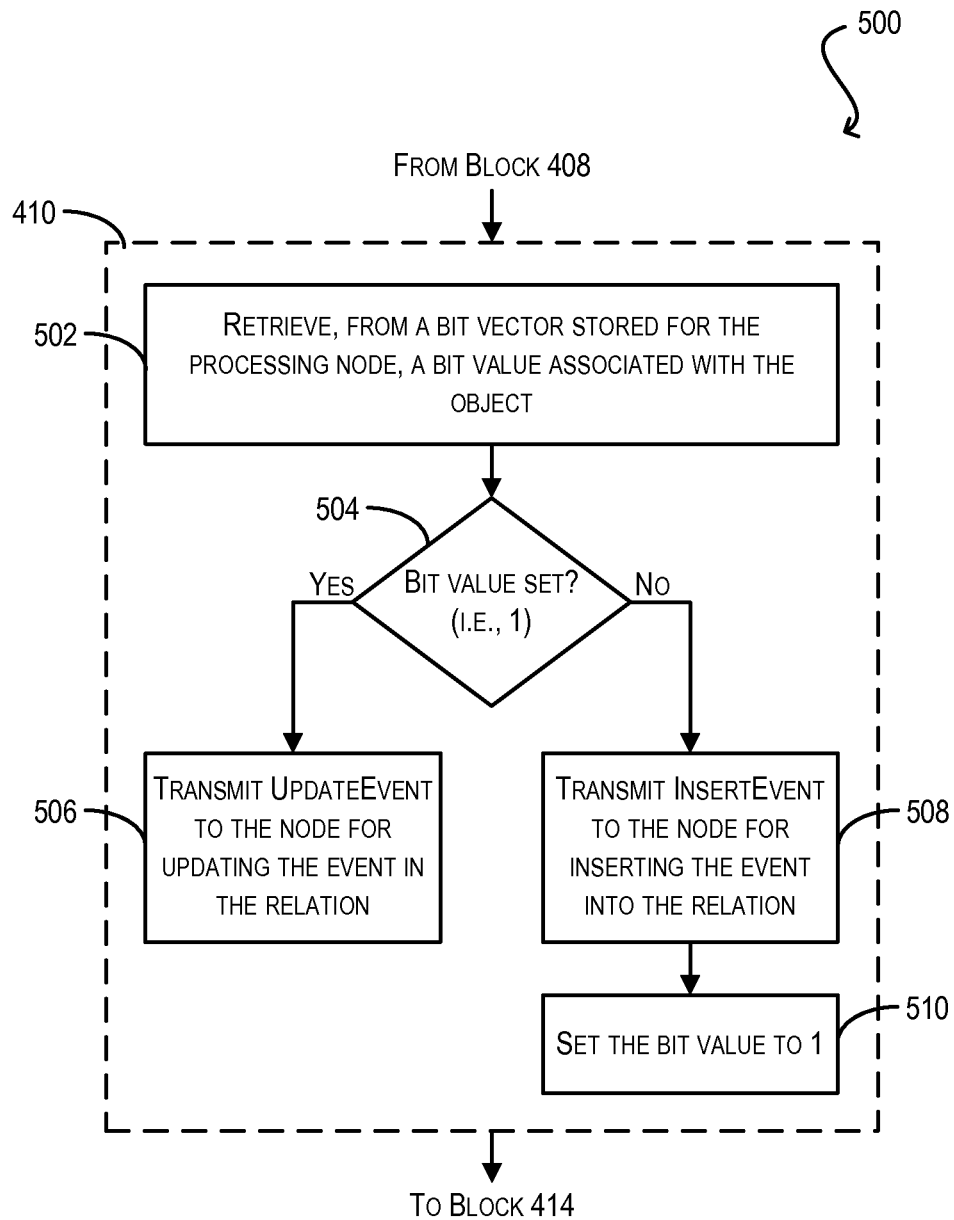Final Office Action for U.S. Appl. No. 13/770,961 dated Aug. 31, 2015, 28 pages.
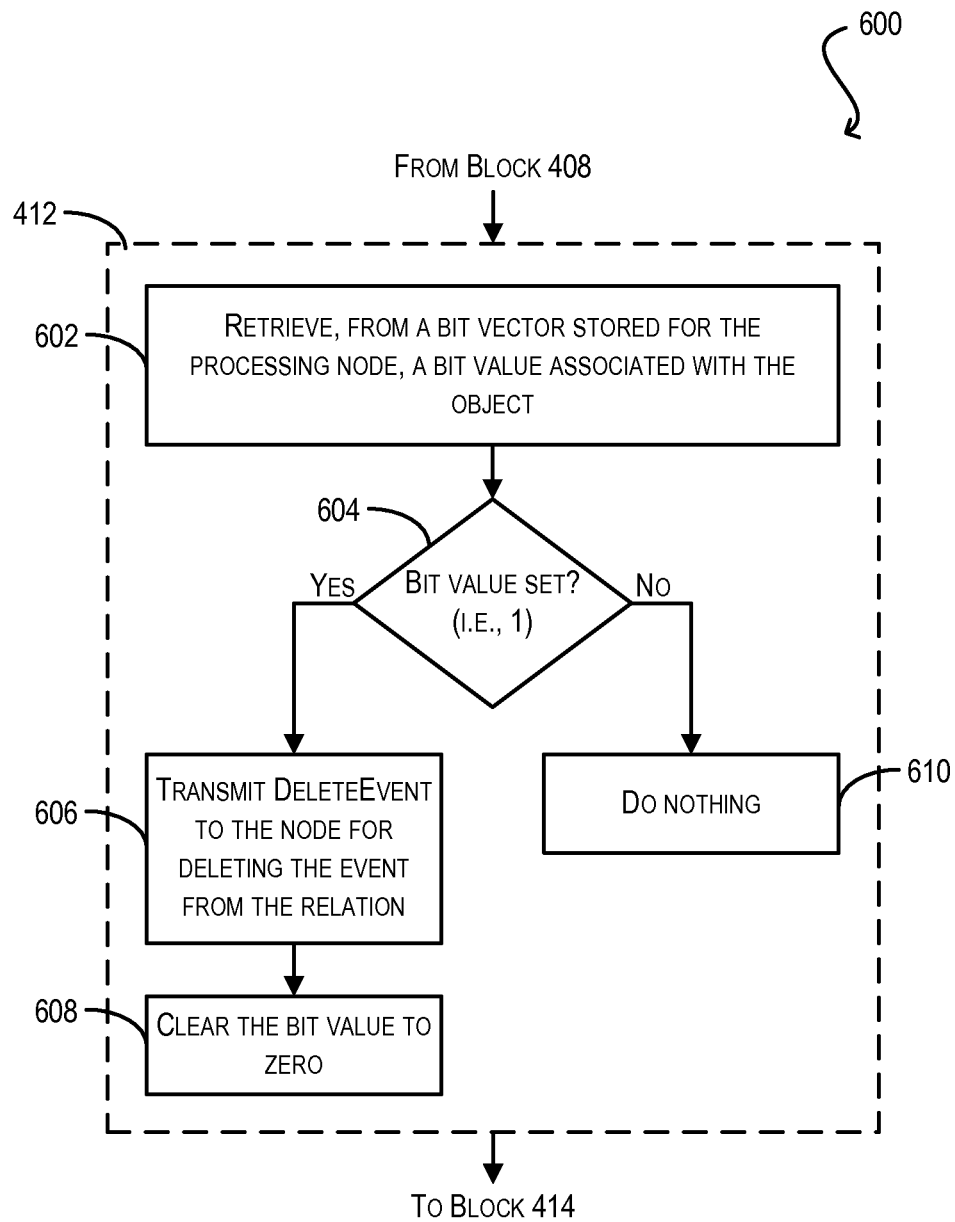
* cited by examiner

FIG. 1

FIG. 2

300

START

302 — RECEIVE AN INPUT EVENT STREAM COMPRISING A SEQUENCE OF EVENTS, THE EVENTS REPRESENTING THE MOVEMENT OF A PLURALITY OF OBJECTS

304 — PARTITION THE INPUT EVENT STREAM AMONG A PLURALITY OF PROCESSING NODES, WHERE EACH NODE IS CONFIGURED TO TRACK OBJECTS WITHIN A PREDEFINED SPATIAL REGION, AND WHERE THE SPATIAL REGIONS FOR AT LEAST TWO PROCESSING NODES OVERLAP

END

FIG. 3

400

FROM BLOCK 302

304

402 — FOR EACH EVENT:

404 — DETERMINE THE PARTICIPATING PROCESSING NODES FOR THE OBJECT IDENTIFIED BY THE EVENT

406 — FOR EACH PROCESSING NODE:

408 — IS THE NODE A PARTICIPATING NODE?

YES   NO

410 — DETERMINE WHETHER TO INSERT OR UPDATE THE EVENT IN THE RELATION OPERATED ON BY THE NODE

412 — DETERMINE WHETHER TO DELETE THE EVENT FROM THE RELATION OPERATED ON BY THE NODE

414 — END NODE FOR LOOP

416 — END EVENT FOR LOOP

FIG. 4

500

FROM BLOCK 408

410

502 — RETRIEVE, FROM A BIT VECTOR STORED FOR THE PROCESSING NODE, A BIT VALUE ASSOCIATED WITH THE OBJECT

504 — BIT VALUE SET? (I.E., 1)

YES

NO

506 — TRANSMIT UPDATEEVENT TO THE NODE FOR UPDATING THE EVENT IN THE RELATION

508 — TRANSMIT INSERTEVENT TO THE NODE FOR INSERTING THE EVENT INTO THE RELATION

510 — SET THE BIT VALUE TO 1

TO BLOCK 414

FIG. 5

600

FROM BLOCK 408

412

602 — RETRIEVE, FROM A BIT VECTOR STORED FOR THE PROCESSING NODE, A BIT VALUE ASSOCIATED WITH THE OBJECT

604 — BIT VALUE SET? (I.E., 1)

YES

NO

606 — TRANSMIT DeleteEvent TO THE NODE FOR DELETING THE EVENT FROM THE RELATION

DO NOTHING — 610

608 — CLEAR THE BIT VALUE TO ZERO

TO BLOCK 414

FIG. 6

702

704

706

708

700

714

DATABASES

710

NETWORK
712

FIG. 7

800

810

COMPUTER
READABLE STORAGE
MEDIA

802

CPU(S)

804

INPUT
DEVICE(S)

806

OUTPUT
DEVICE(S)

808

STORAGE
DEVICE(S)

COMPUTER
READABLE STORAGE
MEDIA READER

812

824

COMMUNICATIONS
SYSTEM

814

PROCESSING
ACCELERATION

816

WORKING
MEMORY

OPERATING
SYSTEM

820

OTHER CODE
(PROGRAMS)

822

818

FIG. 8

TRACKING LARGE NUMBERS OF MOVING OBJECTS IN AN EVENT PROCESSING SYSTEM

CROSS-REFERENCES TO RELATED APPLICATIONS

The present application is related to U.S. patent application Ser. No. 12/949,081, filed Nov. 18, 2010, titled "SPATIAL DATA CARTRIDGE FOR EVENT PROCESSING SYSTEMS," the entire contents of which are incorporated herein by reference for all purposes.

BACKGROUND

Embodiments of the present invention relate in general to event processing, and in particular to techniques for tracking large numbers of moving objects in an event processing system.

Traditional database management systems (DBMSs) execute queries in a "one-off" fashion over finite, stored data sets. For example, a traditional DBMS will receive a request to execute a query from a client, execute the query exactly once against one or more stored database tables, and return a result set to the client.

In recent years, event processing systems have been developed that can execute queries over streams of data rather than finite data sets. Since these streams (referred to herein as "event streams") can comprise a potentially unbounded sequence of input events, an event processing system can execute a query over the streams in a continuous (rather than one-off) manner. This allows the system to continually process new events as they are received. Based on this processing, the event processing system can provide an ongoing stream of results to a client. One example of such an event processing system is the Oracle Complex Event Processing (CEP) Server developed by Oracle Corporation.

Given their unique capabilities, event processing systems are well-suited for enabling applications that require real-time or near real-time processing of streaming data. For instance, event processing systems are particularly well-suited for building "spatial" applications (i.e., applications that require analysis of streams of spatial or geographic location data). Examples of such spatial applications include geographic information systems (GIS), location-enabled business intelligence solutions, geomatics/telematics applications, and the like. Some event processing systems, such as the Oracle CEP Server, provide an extension mechanism for supporting specific spatial features/operations (e.g., spatial data indexing, proximity and overlap determinations, etc.). Information regarding such an extension mechanism can be found in U.S. patent application Ser. No. 12/949,081, filed Nov. 18, 2010, titled "SPATIAL DATA CARTRIDGE FOR EVENT PROCESSING SYSTEMS," the entire contents of which are incorporated herein by reference for all purposes.

One limitation with existing event processing systems that allow spatial operations is that they generally cannot support the tracking of a very large number (e.g., greater than one million) of moving geometries or objects. For example, consider use cases from the telematics market where an application needs to (1) determine all of the vehicles impacted by certain traffic events, or (2) detect "buddies" close to a moving vehicle position, where there is an m to n relation between the number of vehicles and buddies using other vehicles. If the total number of vehicles in these use cases is in the range

of millions, a conventional event processing system generally cannot index and keep track of all of the vehicles in an efficient manner.

BRIEF SUMMARY

Embodiments of the present invention provide techniques for tracking large numbers of moving objects in an event processing system. In one set of embodiments, an input event stream can be received, where the events in the input event stream represent the movement of a plurality of geometries or objects. The input event stream can then be partitioned among a number of processing nodes of the event processing system, thereby enabling parallel processing of one or more continuous queries for tracking the objects. In a particular embodiment, the partitioning can be performed such that (1) each processing node is configured to track objects in a predefined spatial region, and (2) the spatial regions for at least two nodes overlap. This overlapping window enables a single node to find, e.g., all of the objects within a particular distance of a target object, even if the target object is in the process of moving from the region of that node to the overlapping region of another node.

According to one embodiment of the present invention, a method is provided that includes receiving, by a computer system, an input event stream comprising a sequence of events, the sequence of events representing the movement of a plurality of objects. The method further includes partitioning, by the computer system, the input event stream among a plurality of processing nodes to facilitate parallel tracking of the objects, where each processing node is configured to track objects in a predefined spatial region, and where the predefined spatial regions for at least two processing nodes in the plurality of processing nodes overlap.

In one embodiment, each event includes an identifier of an object and a current position of the object.

In one embodiment, partitioning the input event stream includes, for each event, determining a subset of processing nodes in the plurality of processing nodes configured to track objects in a predefined spatial region that encompasses the current position of the object; and for each processing node in the plurality of processing nodes: determining whether the processing node is in the subset; if the processing node is in the subset, determining whether to insert or update the event in a relation operated on by the processing node; and if the processing node is not in the subset, determining whether to delete the event from the relation operated on by the processing node.

In one embodiment, determining whether to insert or update the event in the relation operated on by the processing node includes retrieving, from a bit vector stored for the processing node, a bit value associated with the object; if the bit value is zero, transmitting to the processing node a command for inserting the event into the relation and setting the bit value to one; and if the bit value is one, transmitting to the processing node a command for updating the event in the stream.

In one embodiment, determining whether to delete the event from the relation operated on by the processing node includes retrieving, from a bit vector stored for the processing node, a bit value associated with the object; and if the bit value is one, transmitting to the processing node a command for deleting the event from the relation and clearing the bit value to zero.

In one embodiment, the predefined spatial regions for the plurality of processing nodes are indexed using an R-tree index.

In one embodiment, determining the subset of processing nodes includes performing, based on the current position of the object, a search into the R-tree index.

In one embodiment, the computer system is a load balancing node of an event processing system.

In one embodiment, the sequence of events represent the movement of more than one million distinct objects.

In one embodiment, the plurality of objects are motor vehicles.

In one embodiment, the predefined spatial regions for the plurality of processing nodes are one-dimensional, two-dimensional, or three-dimensional regions.

According to another embodiment of the present invention, a non-transitory computer readable medium having stored thereon program code executable by a processor is provided. The program code includes code that causes the processor to receive an input event stream comprising a sequence of events, the sequence of events representing the movement of a plurality of objects; and code that causes the processor to partition the input event stream among a plurality of processing nodes to facilitate parallel tracking of the objects, where each processing node is configured to track objects in a predefined spatial region, and where the predefined spatial regions for at least two processing nodes in the plurality of processing nodes overlap.

According to another embodiment of the present invention, an event processing system that comprises a load balancer node and a plurality of processing nodes. The load balance node is configured to receive an input event stream comprising a sequence of events, the sequence of events representing the movement of a plurality of objects; and partition the input event stream among the plurality of processing nodes to facilitate parallel tracking of the objects, wherein each processing node is configured to track objects in a predefined spatial region, and wherein the predefined spatial regions for at least two processing nodes in the plurality of processing nodes overlap.

The foregoing, together with other features and embodiments, will become more apparent when referring to the following specification, claims, and accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a simplified block diagram of an event processing system in accordance with an embodiment of the present invention.

FIG. 2 is a simplified block diagram of a load balancing node in accordance with an embodiment of the present invention.

FIGS. 3-6 are flow diagrams of a process for partitioning an input event stream among a plurality of processing nodes in accordance with an embodiment of the present invention.

FIG. 7 is a simplified block diagram of a system environment in accordance with an embodiment of the present invention.

FIG. 8 is a simplified block diagram of a computer system in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION

In the following description, for the purposes of explanation, numerous details are set forth in order to provide an understanding of embodiments of the present invention. It will be apparent, however, to one of ordinary skill in the art that certain embodiments can be practiced without some of these details.

Embodiments of the present invention provide techniques for tracking large numbers of moving objects in an event processing system. In one set of embodiments, an input event stream can be received, where the events in the input event stream represent the movement of a plurality of geometries or objects. The input event stream can then be partitioned among a number of processing nodes of the event processing system, thereby enabling parallel processing of one or more continuous queries for tracking the objects. In a particular embodiment, the partitioning can be performed such that (1) each processing node is configured to track objects in a predefined spatial region, and (2) the spatial regions for at least two nodes overlap. This overlapping window enables a single node to find, e.g., all of the objects within a particular distance of a target object, even if the target object is in the process of moving from the region of that node to the overlapping region of another node.

FIG. 1 is a simplified block diagram of an event processing system 100 according to an embodiment of the present invention. Event processing system 100 can be implemented in hardware, software, or a combination thereof. Unlike traditional DBMSs, event processing system 100 can process queries (i.e., "continuous queries") in a continuous manner over potentially unbounded, real-time event streams. For example, event processing system 100 can receive one or more input event streams from a source (e.g., source 102), execute continuous queries against the input event streams, and generate one or more output event streams destined for a client (e.g., application 104). In a particular embodiment, event processing system 100 can include a mechanism (such as the spatial data cartridge described in U.S. patent application Ser. No. 12/949,081 titled "SPATIAL DATA CARTRIDGE FOR EVENT PROCESSING SYSTEMS") that enables the system to process continuous queries that reference spatial data types, method, fields, and the like.

As shown, event processing system 100 can include a load balancing node 106 and one or more processing nodes 108-112. Although only a single load balancing node and three processing nodes are depicted in FIG. 1, any number of such nodes can be supported.

In one set of embodiments, load balancing node 106 can be configured to partition an input event stream received from source 102 among processing nodes 108-112, thereby enabling the processing nodes to execute one or more continuous queries over the event stream in parallel. By way of example, if the input event stream comprises events E1 through E9, load balancing node 106 might decide to partition the stream such that events E1-E3 are handled by processing node 108, events E4-E6 are handled by processing node 110, and events E7-E9 are handled by processing node 112. In one embodiment, this partitioning can be accomplished by inserting, updating, or deleting events into/from relations maintained by each processing node.

In the context of a spatial application, the input event stream received by load balancing node 106 from source 102 can include events that correspond to the movement of a plurality of geometries or objects (e.g., people, motor vehicles, airplanes, etc.). In these embodiments, load balancing node 106 can partition the events among processing nodes 108-112 based on location information, such that each processing node is responsible for executing queries against a relation representing a predefined spatial region. In various embodiments, the predefined spatial region can be a one-dimensional, two-dimensional, or three-dimensional region. If the spatial application simply requires the identification of non-moving objects in an area of interest (e.g., a geo-fencing use case), the spatial regions handled by each processing node

can be disjoint, and no special processing needs to be performed by load balancing node **106** to insert/update/delete events into the relations associated with the processing nodes—the relations will generally be static.

However, if the spatial application requires the tracking of moving objects across an area of interest, the spatial regions handled by adjacent processing nodes can overlap to some extent. This overlapping window enables a single processing node to find, e.g., all of the objects within a particular distance of a target object, even if the target object is in the process of moving from the region of that node to the overlapping region of another node. The processing performed by load balancing node **106** to enable partitioning across overlapping regions is described in greater detail below.

As described above, processing nodes **108-112** can each be configured to execute one more continuous queries over some partition or subset of the input event stream received from source **102**. In the spatial context, processing nodes **108-112** can each be configured to execute one more continuous queries with respect to objects located in a predefined spatial region. Further, to accommodate the tracking of moving objects, the spatial regions for two more processing nodes can overlap. In one embodiment, processing nodes **108-112** can each correspond to a separate processor in a single machine. In other embodiments, processing nodes **108-112** can each correspond to an event processing server instance running on a separate machine.

It should be appreciated that event processing system **100** of FIG. **1** is illustrative and not intended to limit embodiments of the present invention. For example, event processing system **100** can have other capabilities or include other components that are not specifically described. One of ordinary skill in the art will recognize many variations, modifications, and alternatives.

FIG. **2** is a simplified block diagram that illustrates a functional representation of load balancing node **106** according to an embodiment of the present invention. As shown, load balancing node **106** can including an overlapping partition adapter **200** and a sparse partitioner **202**.

In various embodiments, overlapping partition adapter **200** is configured to receive input events from source **102** and efficiently partition the events among processing nodes **108-112** in a manner that takes into account overlapping regions between the processing nodes. By way of example, consider an object moving across a 2D area, where a first portion of the area is handled by processing node **108** and a second, overlapping portion of the area is handled by processing node **110**. Assume that the object starts out at time T1 within the region handled by processing node **108**, and at time T2 moves into the overlap area between node **108** and node **110**. When this occurs, the event corresponding to the object should be inserted into the relation maintained by processing node **110** (so that it is "visible" to processing node **110**), while also being updating in the relation maintained by processing node **108**. Further, assume that the object moves at time T3 entirely into the region handled by node **110**. At this point, the event corresponding to the object should be deleted from the relation maintained by node **108** while be updated in the relation maintained by node **110**.

To accomplish the above, overlapping partition adapter **200** can carry out an algorithm in load balancing node **106** that appropriately inserts, updates, or deletes events to/from the relations maintained by processing nodes **108-112** to ensure that the processing nodes are correctly updated to track the movement of objects across the nodes. In certain cases, this algorithm can cause an event corresponding to an object to be inserted/updated in the relations of two or more

processing nodes (if it is determined that the object is in an overlapping area between the nodes).

In a particular embodiment, overlapping partition adapter **200** can maintain a bit vector for each processing node, where each bit vector includes a bit entry for each unique object being processing by system **100**. If the bit entry for a given object is set, that indicates that an event corresponding to the object was previously inserted into the relation being handled by the processing node (and it is still there). If the bit entry is not set, that indicates that an event corresponding to the object has not yet been inserted into (or was deleted from) the relation being handled by the processing node. These bit vectors allow overlapping partition adapter **200** to keep track of which processing nodes it has inserted events into, and which processing nodes it needs to update or delete a given event/object from. The details of the algorithm performed by overlapping partition adapter **200** (and how it updates these bit vectors) is described with respect to FIGS. **3-6** below.

Sparse partitioner **202** is an auxiliary component of load balancing node **106** that is configured to identify "participating" processing nodes for a given input event/object. In other words, sparse partitioner **202** can determine which processing nodes handle a spatial region that covers the current location of a given object. In various embodiments, overlapping partition adapter **200** can invoke sparse partitioner **202** to obtain a list of participating processing nodes for each input event or object and use the list within its partitioning algorithm.

In one set of embodiments, sparse partitioner **202** can maintain an Rtree index that indexes bounding rectangles associated with the processing nodes. Each bounding rectangle can represent the spatial region handled by a particular node. Accordingly, when an input event is received, sparse partitioner **202** can use the coordinates for the object associated with the event to perform a search into the Rtree index and return a list or array of processing nodes whose bounding rectangle covers the coordinates.

It should be appreciated that load balancing node **106** of FIG. **2** is illustrative and not intended to limit embodiments of the present invention. For example, load balancing node **106** can have other capabilities or include other components that are not specifically described. One of ordinary skill in the art will recognize many variations, modifications, and alternatives.

FIG. **3** is a flow diagram illustrating a process **300** for partitioning an input event stream among a plurality of processing nodes according to an embodiment of the present invention. In one set of embodiments, process **300** can be carried out by overlapping partition adapter **200** of FIG. **2**. Process **300** can be implemented in hardware, software, or a combination thereof. As software, process **300** can be encoded as program code stored on a machine-readable storage medium.

At block **302**, overlapping partition adapter **200** can receive an input event stream comprising a sequence of events, where the events represent the movement of a plurality of objects. For example, each event can include an identifier of an object, a current position (e.g., coordinates) of the object, and a timestamp. In a particular embodiment, the events in the event stream can represent the movement of a very large number of objects (e.g., greater than one million).

At block **304**, overlapping partition adapter **200** can partition the input event stream among a plurality of processing nodes (e.g., nodes **108-112** of FIG. **1**), where each node is configured to track objects within a predefined spatial region, and where the spatial regions for at least two processing nodes overlap. As discussed above, this overlap enables a single node to find, e.g., all of the objects within a particular distance

of a target object, even if the target object is in the process of moving from the region of that node to the overlapping region of another node.

FIG. 4 illustrates a flow 400 that can be executed by overlapping partition adapter 200 as part of the processing of block 304 of FIG. 3. As shown in FIG. 4, for each event received in the event stream, overlapping partition adapter 200 can determine a list of participating processing nodes for the object identified in the event (blocks 402, 404). As discussed above, this determination can be carried out by passing the position of the object to sparse partitioner 202 of FIG. 2. Sparse partitioner 202 can then use the object's position to perform a search (e.g., an Rtree index search) of processing nodes whose spatial region covers the object's position.

Upon receiving the list of participating processing nodes from sparse partitioner 202, overlapping partition adapter 200 can iterate through all of the processing nodes in the system and determine whether a given node is a participating node (e.g., is in the list returned by sparse partitioner 202) (blocks 406, 408). If a given node is a participating node, that means the object identified by the current event should be tracked by the node. Accordingly, overlapping partition adapter 200 can determine whether to insert or update the event into the relation maintained by the node (block 410). If the node is not a participating node, that means the object identified by the event should not (or should no longer) be tracked by the node. Accordingly, overlapping partition adapter 200 can determine whether to delete the event from the relation maintained by the node (block 412).

Once the determination at block 410 or 412 is made, overlapping partition adapter 200 can continue to iterate through all of the processing nodes, and repeat this loop for each incoming event (blocks 414, 416).

FIG. 5 illustrates a flow 500 that can be executed by overlapping partition adapter 200 as part of the processing of block 410 of FIG. 4. At block 502, overlapping partition adapter 200 can retrieve, from a bit vector stored for the current processing node, a bit value associated with the current object. As discussed above with respect to FIG. 2, a bit vector is stored for each processing node in the system and reflects which objects are currently being tracked by the node.

If the bit value for the object is set (i.e., has a value of one), overlapping partition adapter 200 can transmit an updateevent command to the processing node for updating the event in the relation (blocks 504, 506). If the bit value for the object is not set (i.e., has a value of zero), overlapping partition adapter 200 can transmit an insertevent command to the processing node for inserting the event into the relation (blocks 504, 508). Adapter 200 can then set the bit value (i.e., change the value to one) to indicate that the processing node is now tracking the object (block 510).

FIG. 6 illustrates a flow 600 that can be executed by overlapping partition adapter 200 as part of the processing of block 412 of FIG. 4. Like block 502 of FIG. 5, overlapping partition adapter 200 can retrieve, from a bit vector stored for the current processing node, a bit value associated with the current object (block 602). If the bit value for the object is set (i.e., has a value of one), overlapping partition adapter 200 can transmit a deleteevent command to the processing node for deleting the event in the relation (blocks 604, 606). The adapter can then clear the bit value (i.e., change the value to zero) to indicate that the processing node is no longer tracking the object (block 608). If the bit value for the object is not set (i.e., has a value of zero), overlapping partition adapter 200 can do nothing (block 610).

It should be appreciated that the flow diagrams depicted in FIGS. 3-6 are illustrative and that variations and modifica-

tions are possible. Steps described as sequential can be executed in parallel, order of steps can be varied, and steps can be modified, combined, added, or omitted. One of ordinary skill in the art would recognize many variations, modifications, and alternatives.

Using the techniques described above, embodiments of the present invention can support very large scale moving object tracking in an event processing system (e.g., greater than one million objects), while using a relatively small amount of working memory. For example, only 128 Kilobytes of memory are needed per processing node (for the bit vector) for handling one million unique moving objects. Further, note that the module for identifying participating nodes (i.e., sparse partitioner 202) is separate from the insert/update/delete event processing performed by overlapping partition adapter 200. Accordingly different types of partitioning policies can be plugged into the system to support different spatial use cases.

FIG. 7 is a simplified block diagram illustrating a system environment 700 that can be used in accordance with an embodiment of the present invention. As shown, system environment 700 can include one or more client computing devices 702, 704, 706, 708, which can be configured to operate a client application such as a web browser, a UNIX/SOLARIS terminal application, and/or the like. In one set of embodiments, client computing devices 702, 704, 706, 708 may be configured to run one or more client applications that interact with event processing system 100 of FIG. 1.

Client computing devices 702, 704, 706, 708 can be general purpose personal computers (e.g., personal computers and/or laptop computers running various versions of MICROSOFT WINDOWS and/or APPLE MACINTOSH operating systems), cell phones or PDAs (running software such as Microsoft Windows Mobile and being Internet, e-mail, SMS, Blackberry, or other communication protocol enabled), and/or workstation computers running any of a variety of commercially-available UNIX or UNIX-like operating systems (including without limitation the variety of GNU/Linux operating systems). Alternatively, client computing devices 702, 704, 706, 708 can be any other electronic device capable of communicating over a network, such as network 712 described below. Although system environment 700 is shown with four client computing devices, it should be appreciated that any number of client computing devices can be supported.

System environment 700 can further include a network 712. Network 712 can be any type of network familiar to those skilled in the art that can support data communications using a network protocol, such as TCP/IP, SNA, IPX, AppleTalk, and the like. Merely by way of example, network 712 can be a local area network (LAN), such as an Ethernet network, a Token-Ring network and/or the like; a wide-area network; a virtual network, including without limitation a virtual private network (VPN); the Internet; an intranet; an extranet; a public switched telephone network (PSTN); an infra-red network; a wireless network (e.g., a network operating under any of the IEEE 802.11 suite of protocols, the Bluetooth protocol known in the art, and/or any other wireless protocol); and/or any combination of these and/or other networks.

System environment 700 can further include one or more server computers 710 which can be general purpose computers, specialized server computers (including, e.g., PC servers, UNIX servers, mid-range servers, mainframe computers, rack-mounted servers, etc.), server farms, server clusters, or any other appropriate arrangement and/or combination. Server 710 can run an operating system including any of those

discussed above, as well as any commercially available server operating system. Server **710** can also run any of a variety of server applications and/or mid-tier applications, including web servers, FTP servers, CGI servers, JAVA virtual machines, and the like. In one set of embodiments, server **710** may correspond to a machine configured to run event processing system **100** of FIG. **1**.

System environment **700** can further include one or more databases **714**. In one set of embodiments, databases **714** can include databases that are managed by server **710** (e.g., database **108** of FIG. **1**). Databases **714** can reside in a variety of locations. By way of example, databases **714** can reside on a storage medium local to (and/or resident in) one or more of computers **702**, **704**, **706**, **708**, and **710**. Alternatively, databases **714** can be remote from any or all of computers **702**, **704**, **706**, **708**, and **710**, and/or in communication (e.g., via network **712**) with one or more of these. In one set of embodiments, databases **714** can reside in a storage-area network (SAN) familiar to those skilled in the art.

FIG. **8** is a simplified block diagram illustrating a computer system **800** that can be used in accordance with an embodiment of the present invention. In various embodiments, computer system **800** can be used to implement any of computers **702**, **704**, **706**, **708**, and **710** described with respect to system environment **700** above. As shown, computer system **800** can include hardware elements that are electrically coupled via a bus **824**. The hardware elements can include one or more central processing units (CPUs) **802**, one or more input devices **804** (e.g., a mouse, a keyboard, etc.), and one or more output devices **806** (e.g., a display device, a printer, etc.). Computer system **800** can also include one or more storage devices **808**. By way of example, the storage device(s) **808** can include devices such as disk drives, optical storage devices, and solid-state storage devices such as a random access memory (RAM) and/or a read-only memory (ROM), which can be programmable, flash-updateable and/or the like.

Computer system **800** can additionally include a computer-readable storage media reader **812**, a communications subsystem **814** (e.g., a modem, a network card (wireless or wired), an infra-red communication device, etc.), and working memory **818**, which can include RAM and ROM devices as described above. In some embodiments, computer system **800** can also include a processing acceleration unit **816**, which can include a digital signal processor (DSP), a special-purpose processor, and/or the like.

Computer-readable storage media reader **812** can be connected to a computer-readable storage medium **810**, together (and, optionally, in combination with storage device(s) **808**) comprehensively representing remote, local, fixed, and/or removable storage devices plus storage media for temporarily and/or more permanently containing computer-readable information. Communications system **814** can permit data to be exchanged with network **712** and/or any other computer described above with respect to system environment **700**.

Computer system **800** can also comprise software elements, shown as being currently located within working memory **818**, including an operating system **820** and/or other code **822**, such as an application program (which may be a client application, Web browser, middle tier/server application, etc.). It should be appreciated that alternative embodiments of computer system **800** can have numerous variations from that described above. For example, customized hardware can be used and particular elements can be implemented in hardware, software, or both. Further, connection to other computing devices such as network input/output devices can be employed.

Computer readable storage media for containing code, or portions of code, executable by computer system **800** can include any appropriate media known or used in the art, such as but not limited to volatile/non-volatile and removable/non-removable media. Examples of computer-readable storage media include RAM, ROM, EEPROM, flash memory, CD-ROM, digital versatile disk (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, an any other medium that can be used to store data and/or program code and that can be accessed by a computer.

Although specific embodiments of the invention have been described above, various modifications, alterations, alternative constructions, and equivalents are within the scope of the invention. For example, embodiments of the present invention are not restricted to operation within certain specific data processing environments, but are free to operate within a plurality of data processing environments. Further, although embodiments of the present invention have been described with respect to certain flow diagrams and steps, it should be apparent to those skilled in the art that the scope of the present invention is not limited to the described diagrams/steps.

Yet further, although embodiments of the present invention have been described using a particular combination of hardware and software, it should be recognized that other combinations of hardware and software are also within the scope of the present invention.

The specification and drawings are, accordingly, to be regarded in an illustrative rather than restrictive sense. It will be evident that additions, subtractions, and other modifications may be made thereunto without departing from the broader spirit and scope of the invention as set forth in the following claims.

What is claimed is:

1. A method comprising:

receiving, by a computer system, an input event stream comprising a sequence of events, the sequence of events representing movement of a plurality of objects, wherein the input event stream includes a first event of the sequence of events, the first event being associated with a first object of the plurality of objects; and

partitioning, by the computer system, the input event stream among a plurality of processing nodes to facilitate parallel tracking of the plurality of objects using a plurality of bit vectors that correspond to the plurality of processing nodes, wherein each of the plurality of bit vectors comprises a plurality of bit values corresponding to the plurality of objects, wherein each processing node of the plurality of processing nodes is configured to track those of the plurality of objects having spatial positions in a predefined spatial region, and wherein the predefined spatial regions for at least two processing nodes in the plurality of processing nodes overlap, wherein the partitioning for the first event includes:

determining, by the computer system, a spatial position of the first object based upon the first event;

determining, by the computer system, that the spatial position of the first object is in the predefined spatial region tracked by a first processing node of the plurality of processing nodes;

determining, by the computer system using a first bit value of a first bit vector of the plurality of bit vectors that is associated with the first processing node, that the first processing node is not currently tracking the first object, wherein the first bit value is associated with the first object;

changing, by the computer system, the first bit value from a first value to a second value that is different than the first value; and

transmitting, by the computer system, a command to the first processing node for inserting the first event into a first spatial-region-representing relation that is operated upon by the first processing node.

2. The method of claim 1 wherein:

the plurality of processing nodes operate upon a plurality of spatial-region-representing relations, the plurality of spatial-region-representing relations including the first spatial-region-representing relation;

each event of the sequence of events includes an identifier of the associated object and a current spatial position of the object; and

the partitioning the input event stream comprises, for each received event:

determining a subset of processing nodes in the plurality of processing nodes configured to track objects in the predefined spatial region that encompasses the current position of the object; and

for each processing node in the plurality of processing nodes:

determining whether the processing node is in the subset;

when the processing node is in the subset, determining whether to cause the event to be inserted or updated in the spatial-region-representing relation operated on by the processing node; and

when the processing node is not in the subset, determining whether to cause the event to be deleted from the spatial-region-representing relation operated on by the processing node.

3. The method of claim 2 wherein determining whether to insert or update the event in the relation operated on by the processing node comprises:

retrieving, from the bit vector associated with the processing node, the bit value associated with the object;

when the bit value is the first value:

transmitting to the processing node a command for inserting the event into the spatial-region-representing relation; and

setting the bit value to the second value; and

when the bit value is the second value, transmitting a command to the processing node for updating the event in the spatial-region-representing relation.

4. The method of claim 2 wherein determining whether to delete the event from the relation operated on by the processing node comprises:

retrieving, from the bit vector stored for the processing node, the bit value associated with the object; and

when the bit value is the second value:

transmitting to the processing node a command for deleting the event from the relation; and

setting the bit value to the first value.

5. The method of claim 2 wherein:

the predefined spatial regions for the plurality of processing nodes are indexed using an R-tree index; and

the determining the subset of processing nodes comprises performing, based on the current position of the object, a search into the R-tree index.

6. The method of claim 1 wherein the computer system is a load balancing node of an event processing system.

7. The method of claim 1 wherein the sequence of events represent the movement of more than one million distinct objects.

8. The method of claim 1 wherein the plurality of objects are motor vehicles.

9. The method of claim 1 wherein the predefined spatial regions for the plurality of processing nodes are one-dimensional, two-dimensional, or three-dimensional regions.

10. The method of claim 1, wherein the input event stream further includes a second event of the sequence of events that is associated with a second object of the plurality of objects, and wherein the partitioning for the second event includes:

determining, by the computer system, a second spatial position of the second object based upon the second event;

determining, by the computer system, that the second spatial position of the second object is in a second predefined spatial region tracked by a second processing node of the plurality of processing nodes and a third predefined spatial region tracked by a third processing node of the plurality of processing nodes;

determining, by the computer system using a second bit value of a second bit vector of the plurality of bit vectors that is associated with the second processing node, that the second processing node is currently tracking the second object, wherein the second bit value is associated with the second object;

transmitting, by the computer system, a second command to the second processing node for updating the second event in a second spatial-region-representing relation that is operated upon by the second processing node;

determining, by the computer system using a third bit value of a third bit vector of the plurality of bit vectors that is associated with the third processing node, that the third processing node is not currently tracking the second object, wherein the third bit value is associated with the second object;

changing, by the computer system, the third bit value from the first value to the second value; and

transmitting, by the computer system, a third command to the third processing node for inserting the second event into a third spatial-region-representing relation that is operated upon by the third processing node.

11. The method of claim 1, wherein the partitioning for the first event further includes:

prior to the changing the first bit value, identifying, by the computer system, from the first bit vector corresponding to the first processing node, the first bit value;

determining, by the computer system, a type of the command to be sent to the first processing node based on the first bit value.

12. The method of claim 1, wherein the input event stream further includes a second event of the sequence of events that is also associated with the first object, wherein the partitioning further includes:

determining, by the computer system, a second spatial position of the first object based upon the second event;

determining, by the computer system, that the second spatial position of the second object is not in the first predefined spatial region tracked by the first processing node;

determining, by the computer system using a second bit value of a first bit vector, that the first processing node is currently tracking the first object;

changing, by the computer system, the first bit value from the second value to the first value; and

transmitting, by the computer system, another command to the first processing node for deleting one or more events

associated with the first object from the first spatial-region-representing relation operated upon by the first processing node.

13. A non-transitory computer readable medium having stored thereon program code executable by a processor, the program code comprising:

code that causes the processor to receive an input event stream comprising a sequence of events, the sequence of events representing movement of a plurality of objects, wherein the input event stream includes a first event of the sequence of events, the first event being associated with a first object of the plurality of objects; and

code that causes the processor to partition the input event stream among a plurality of processing nodes to facilitate parallel tracking of the plurality of objects using a plurality of bit vectors that correspond to the plurality of processing nodes, wherein each of the plurality of bit vectors comprises a plurality of bit values corresponding to the plurality of objects, wherein each processing node of the plurality of processing nodes is configured to track those of the plurality of objects having spatial positions in a predefined spatial region, and wherein the predefined spatial regions for at least two processing nodes in the plurality of processing nodes overlap, wherein the partitioning for the first event includes:

determining a spatial position of the first object based upon the first event;

determining that the spatial position of the first object is in the predefined spatial region tracked by a first processing node of the plurality of processing nodes;

determining, using a first bit value of a first bit vector of the plurality of bit vectors that is associated with the first processing node, that the first processing node is not currently tracking the first object, wherein the first bit value is associated with the first object;

changing the first bit value from a first value to a second value that is different than the first value; and

transmitting a command to the first processing node for inserting the first event into a first spatial-region-representing relation that is operated upon by the first processing node.

14. The non-transitory computer readable medium of claim 13, wherein:

the plurality of processing nodes operate upon a plurality of spatial-region-representing relations, the plurality of spatial-region-representing relations including the first spatial-region-representing relation;

each event of the sequence of events includes an identifier of the associated object and a current spatial position of the object; and

the partitioning the input event stream comprises, for each received event:

determining a subset of processing nodes in the plurality of processing nodes configured to track objects in the predefined spatial region that encompasses the current position of the object; and

for each processing node in the plurality of processing nodes:

determining whether the processing node is in the subset;

when the processing node is in the subset, determining whether to cause the event to be inserted or updated in the spatial-region-representing relation operated on by the processing node; and

when the processing node is not in the subset, determining whether to cause the event to be deleted

from the spatial-region-representing relation operated on by the processing node.

15. The non-transitory computer readable medium of claim 13, wherein the partitioning for the first event further includes:

prior to the changing the first bit value, identifying, from the first bit vector corresponding to the first processing node, the first bit value;

determining a type of the command to be sent to the first processing node based on the first bit value.

16. The non-transitory computer readable medium of claim 13, wherein the input event stream further includes a second event of the sequence of events that is also associated with the first object, wherein the partitioning further includes:

determining a second spatial position of the first object based upon the second event;

determining that the second spatial position of the second object is not in the first predefined spatial region tracked by the first processing node;

determining, using a second bit value of a first bit vector, that the first processing node is currently tracking the first object;

changing the first bit value from the second value to the first value; and

transmitting another command to the first processing node for deleting one or more events associated with the first object from the first spatial-region-representing relation operated upon by the first processing node.

17. An event processing system, comprising:

a load balancer node; and

a plurality of processing nodes,

wherein the load balancer node is configured to:

receive an input event stream comprising a sequence of events, the sequence of events representing movement of a plurality of objects, wherein the input event stream includes a first event of the sequence of events, the first event being associated with a first object of the plurality of objects; and

partition the input event stream among the plurality of processing nodes to facilitate parallel tracking of the plurality of objects using a plurality of bit vectors that correspond to the plurality of processing nodes, wherein each of the plurality of bit vectors comprises a plurality of bit values corresponding to the plurality of objects, wherein each processing node of the plurality of processing nodes is configured to track those of the plurality of objects having spatial positions in a predefined spatial region, and wherein the predefined spatial regions for at least two processing nodes in the plurality of processing nodes overlap, wherein the partitioning for the first event includes:

determining a spatial position of the first object based upon the first event;

determining that the spatial position of the first object is in the predefined spatial region tracked by a first processing node of the plurality of processing nodes;

determining, using a first bit value of a first bit vector of the plurality of bit vectors that is associated with the first processing node, that the first processing node is not currently tracking the first object, wherein the first bit value is associated with the first object;

changing the first bit value from a first value to a second value that is different than the first value; and

transmitting a command to the first processing node for inserting the first event into a first spatial-region-representing relation that is operated upon by the first processing node.

15

18. The event processing system of claim **17**, wherein:

the plurality of processing nodes operate upon a plurality of spatial-region-representing relations, the plurality of spatial-region-representing relations including the first spatial-region-representing relation;

each event of the sequence of events includes an identifier of the associated object and a current spatial position of the object; and

the partitioning the input event stream comprises, for each received event:

determining a subset of processing nodes in the plurality of processing nodes configured to track objects in the predefined spatial region that encompasses the current position of the object; and

for each processing node in the plurality of processing nodes:

determining whether the processing node is in the subset;

when the processing node is in the subset, determining whether to cause the event to be inserted or updated in the spatial-region-representing relation operated on by the processing node; and

when the processing node is not in the subset, determining whether to cause the event to be deleted from the spatial-region-representing relation operated on by the processing node.

16

19. The event processing system of claim **17**, wherein the partitioning for the first event further includes:

prior to the changing the first bit value, identifying, from the first bit vector corresponding to the first processing node, the first bit value;

determining a type of the command to be sent to the first processing node based on the first bit value.

20. The event processing system of claim **17**, wherein the input event stream further includes a second event of the sequence of events that is also associated with the first object, wherein the partitioning further includes:

determining a second spatial position of the first object based upon the second event;

determining that the second spatial position of the second object is not in the first predefined spatial region tracked by the first processing node;

determining, using a second bit value of a first bit vector, that the first processing node is currently tracking the first object;

changing the first bit value from the second value to the first value; and

transmitting another command to the first processing node for deleting one or more events associated with the first object from the first spatial-region-representing relation operated upon by the first processing node.

* * * * *

CERTIFICATE OF CORRECTION

PATENT NO. : 9,189,280 B2   Page 1 of 1
APPLICATION NO. : 13/107742
DATED   : November 17, 2015
INVENTOR(S) : Park et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Title page, Item (56)

On page 5, column 1, under other publications, line 34, delete "Postdam," and insert -- Potsdam, --, therefor.

On page 6, column 1, under other publications, line 54, delete "PostgresSQL" and insert
-- PostgreSQL --, therefor.

On page 6, column 1, under other publications, line 54, delete "PostgresSQL 8.2" and insert
-- PostgreSQL 8.2 --, therefor.

On page 6, column 1, under other publications, line 56, delete "PostgresSQL" and insert
-- PostgreSQL --, therefor.

On page 6, column 1, under other publications, line 56, delete "PostgresSQL 8.2" and insert
-- PostgreSQL 8.2 --, therefor.

On page 8, column 2, under other publications, line 36, delete "Integrationand" and insert
-- Integration and --, therefor.

On page 10, column 2, under other publications, line 17, delete "Office office" and insert -- Office --, therefor.

Signed and Sealed this
Sixth Day of September, 2016

Michelle K. Lee

Michelle K. Lee
Director of the United States Patent and Trademark Office